DM 7,- Sfr 7,- Ös 56,- Juni ´88 Nr. 6

AMIGA

KICK START

COMPUTER MAGAZIN



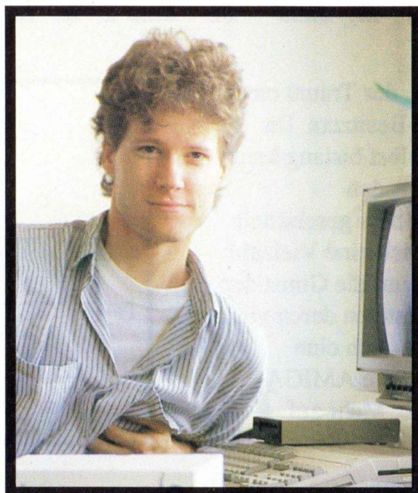
FESTE PLATTELN

Festplatten am AMIGA

GUT GEBACKEN

Die Software des JUNIOR PROMMER

Kickstart/Workbench V.1.3 · Bootletter · DFÜ
DISCovery · Zeichenkurs · LISP · KICKS
Photon Paint · Spiele . Tips & Tricks · TOP 12



Endlich ...

Endlich ist diese Ausgabe der KICKSTART vollendet (und diesmal hoffentlich rechtzeitig am Kiosk!),.. - endlich ein wenig Zeit, um uns der Tatsache bewußt zu werden, daß wir schon seit einem Jahr diese Zeitschrift herausgeben. Denkt man an die Zeit zurück, als die KICKSTART noch hauptsächlich von drei Redakteuren geplant, gestaltet und zum größten Teil auch geschrieben wurde, so hat sich in der Zwischenzeit einiges geändert. Vier Redakteure, knapp 10 redaktionelle Mitarbeiter und eine große Anzahl freier Autoren tragen jeden Monat dazu bei, daß die KICKSTART für jeden etwas interessantes parat hat. Auch Sie -die Leser- haben schon seit Anfang an einen großen Einfluß auf diese Zeitung, denn Ihre Kritik und Anregungen versuchen wir umzusetzen. Und außerdem kommen viele Listings, Tests von Hard- und Software und andere Artikel, aus den Reihen der Leserschaft. Wir möchten Ihnen dafür herzlich danken und Sie auch weiterhin dazu anregen, sich rege zu beteiligen.

Sehr wichtig sind in diesem Zusammenhang auch die Rubriken KICKS, Tips & Tricks und die Leserbriefe. Hier können Sie Ihre Probleme schildern oder anderen AMIGA-Benutzern bei deren Schwierigkeiten helfen, denn fast jeder kommt einmal an einen Punkt, an dem er nicht mehr weiterkommt. Dann ist es von unschätzbarem Vorteil, wenn man eine Hilfestellung erhalten kann und nicht aus Verzweiflung gleich den AMIGA aus dem Fenster werfen will.

Wir hoffen deshalb, daß Sie sich auch im nächsten KICKSTART-Jahr an den Rubriken und Aktionen der Zeitung beteiligen und so dazu beitragen, daß die KICKSTART so wird, wie Sie sich das vorstellen.

Ihr

M. Nerding

Markus Nerding

INHALT

FESTE PLATTELN

AKTUELLES

SOFTWARE

HARDWARE

GRUNDLAGEN

FESTE PLATTELN

Eine Festplatte, der Traum eines jeden AMIGA-Besitzers. Da Commodore selbst bislang keine eigene Platte aus den Entwicklungslabors geschüttelt hat, bemühen sich eine Vielzahl Fremdanbieter um die Gunst der Kunden. Wir testeten deren vier. Dabei fand sich auch eine Premiere - die erste AMIGA-Autoboot-Platte. Mehr auf Seite *10*

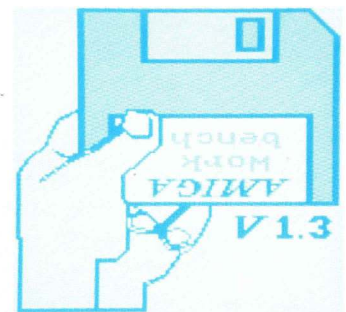


Gut gebacken

Im letzten Heft stellten wir die Hardware des JUNIOR PROMMER vor. Nun präsentieren wir die Software, die den kleinen Allesbrenner zum Leben erweckt. Darauf haben die EPROMS der 27XXX-Serie lange gewartet. Die 'Backrezepte' finden Sie auf Seite *33*

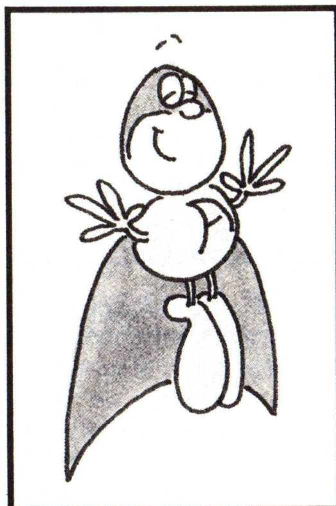
Neue KICKSTART und WORKBENCH

Eine neue KICKSTART? Nichts Besonderes. Die können Sie jeden Monat am Kiosk bekommen. Doch es gibt ja noch ein gleichnamiges Betriebssystem. Wir, die MERLIN'sche KICKSTART, testeten die COMMODORE'sche KICKSTART. Und da wir gerade dabei waren, testeten wir die neue WORKBENCH gleich mit. Was dabei raus kam, erfahren Sie auf Seite *21*



Die Entdeckung

Der AMIGA ist schon eine seltsame Maschine. Mal tut er dies, mal jenes. Der Anwender ist nicht selten von diesem Eigenleben überrascht und muß sich wundern, daß er seinen Computer nicht mehr unter Kontrolle hat. Schuld daran ist AMIGAMAN, der erst kurz vor Redaktionsschluß entdeckte Super-Duper-Held. Er ist das denkende Hirn, der Manager im AMIGA, Bändiger des Gurus, Zöllner im I/O-Bereich, Verfasser des Betriebssystems-Schutzes, Datenbuskontrolleur, Supervisor und Leiter des Meditationszentrums. Bei solch zahlreichen Aufgaben, passieren schon einmal Fehler, doch ein gewisser Stolz ist Ihm trotz allem nicht zu nehmen - dem AMIGAMAN.

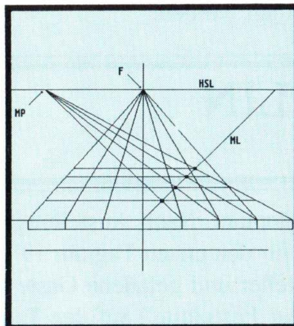


Seite *64, 15, 23*

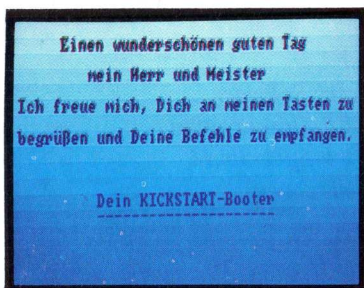
Die Zeichenkiste

Der AMIGA ist für grafische Anwendungen ideal, doch auch bei den besten Zeichenprogrammen braucht man auch ein glückliches Händchen. In diesem Kurs sollen Sie mit einigen grundlegenden Zeichentechniken vertraut gemacht werden. Teil 1 beschäftigt sich mit perspektivischen Konstruktionen.

Seite 88



Bootletter



Wie wäre es damit? Sie schalten Ihren Rechner ein und sofort erscheinet ein farbenfrohes Bild mit einer netten Einschaltmeldung.
Mit Bootletter läßt sich dies leicht bewerkstelligen.

Seite *65*

LISTING

KICKS fÜR INSIDER

MS-DOS

SPIELE

RUBRIKEN

NEWS

KICKSTART 6/88

Wechselplatten-Laufwerk für Amiga

Die englische Firma Burocare Computer Systems bringt für alle Amiga-Modelle eine Wechselfestplatte auf den Markt. Die 5.25"-Platte wird intern in den Amiga 2000 eingesteckt oder extern über einen Adapter an den Modellen A1000 und A500 angeschlossen. Die Platten können beim Amiga 2000 in AmigaDos- und MS-DOS-Format
beschrieben werden, vorausgesetzt, ein XT-Bridgeboard ist vorhanden. Eine Mischung beider Formate ist ebenfalls kein Problem. Das Wechselplatten-System besitzt einige Vorteile gegenüber reinen Festplatten oder anderen Massenspeichern. Beispielsweise können weitere Platten für wenig Geld nachgekauft werden, somit ergibt sich ein geringer Preis für 1 MByte. Die durchschnittliche Zugriffszeit gibt die Firma Burocare Computer mit 10 ms an.

Ein deutscher Preis ist zur Zeit noch nicht kalkuliert, so daß lediglich die englischen Preise vorliegen. Das Wechselplatten-Laufwerk wird mit 1695 Pfund angegeben, jede weitere 20 MByte Wechselplatte schlägt mit 50 Pfund zu Buche. Eine 200 MByte Wechselplatte soll mit 500 Pfund veranschlagt werden.

AMIGA WINDOW BERLIN

1. Deutsche Amiga-Ausstellung

In Berlin findet am Freitag, dem 3. und am Samstag, dem 4. Juni 1988 die erste reine Amiga-Ausstellung in Berlin statt. Die Ziele der AMIGA WINDOW Berlin bestehen in der Weitergabe von Information, in der Präsentation von alten und neuen Produkten von Commodore und anderen Herstellern und im Verkauf von Hardware und Software.

Für den Amiga-Anwender veranstaltet die AMIGA WINDOW eine Reihe von Workshops: Desktop Puplishing, Repro-Grafik und Business-Anwendungen, am Freitag, dem 3.6., vormittags; Schule Universität, Lernen, ebenfalls am Freitag, den 3.6., jedoch nachmittags; Desktop Video und Animation, am Samstag, den 4.6., vormittags; Musik und Sound, Samstag, den 4.6., nachmittags. Des weiteren findet ein weiterer Workshop mindestens 4 x statt: Einführung in Amiga für Einsteiger. Sonderveranstaltungen sind ebenfalls geplant. Nach Austellungsschluß steht für den ersten Tag um 19 Uhr für Aussteller und geladene Gäste "Menü auf der Festplatte" auf der Tagesordnung. Am Sonntag um 11 Uhr findet der Ausklang der AMIGA WINDOW statt, Gäste und Aussteller sind eingeladen.

Für den Amiga-Anwender soll diese reine, vom technicSupport organisierte, Amiga-(Verkaufs-)Messe eine Fülle von Informationen und Neuigkeiten bringen. Man hofft auf einen ebenso großen Erfolg dieser Messe, wie bei ähnlichen Veranstaltungen in Amerika.

In Planung sind weitere Großstädte für selbige Ausstellung. Hoffentlich wirken sich die recht happigen Bedingungen (der Veranstalter ist mit 20% am Umsatz der Aussteller beteiligt) nicht negativ auf die Zahl anwesender Aussteller aus, aber das wird sich zeigen.

CONFERENCE of EUROPEAN PROGRAMMERS (CEP)-

erste Europäische Programmiererkonferenz in Ludwigshafen

Die erste europäische Konferenz für 68.000er-Programmierer wird vom 15. - 17. Juli 1988 in Ludwigshafen stattfinden. Der Schwerpunkt wird dabei sicherlich vom Amiga gesetzt.

Neben den 68.000-Rechnern wird es aber auch Präsentationen für den Archimedes geben.

Dr. Tim King (Entwickler des Amiga-Dos und zur Zeit Mitentwickler des Helios-Betriebssystems für Transputerrechner) hat seine aktive Mitwirkung bereits ebenso zugesagt wie einige bekannten deutschen Programmierer.

Die Teilnehmerzahl für die Hearings und Diskussionen ist auf 250 Personen begrenzt. Der Veranstalter Discovery Software erhebt eine Teilnahmegebühr von 25.- DM für die Gesamtveranstaltung. Daran kann man vor allem erkennen, daß die Veranstaltung nicht als überdimensionale Copyparty gedacht ist.

Wer mehr Informationen haben will, sollte umgehend einen mit 0.80 DM frankierten und mit seiner Adresse versehenen Rückumschlag einsenden.

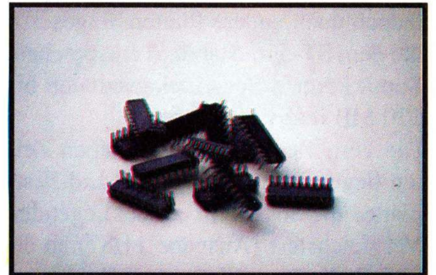
Adresse:
Discovery Software
Abt. CEP-Organisation
Friedrich-Spee-Str.11
8700 Würzburg

Instant MUSIC

Electronic Arts ist vor allem bekannt durch das Zeichenprogrammm Deluxe Paint. Mit Instant Music vertreibt das Softwarehaus seit geraumer Zeit auch ein Musikprogramm, doch dieses war bisher nur mit englischsprachigem Handbuch erhältlich. Ab sofort wird Instant Music mit einem deutschen Handbuch ausgeliefert, so daß die Möglichkeiten des Programms leichter einzusetzen sind. Instant Music besticht durch die extravagante Eingabe des Sounds:
Der Anwender muß keine Noten können, sondern definiert nur einzelne Blöcke im Anzeigefenster. Wie das wohl klingen mag?

RAMs werden immer teurer!

Entgegen allen Erwartungen sind die 256KBit-RAM-Bausteine im Preis erheblich gestiegen. Leider sind dies die Bausteine, die im Amiga für den Speicher und auch in fast allen erhältlichen Speichererweiterungen verwendet werden. Aus diesem Grunde bekommen viele Händler Schwierigkeiten mit ihren Speichererweiterungen; zum einen mit der Auslieferung der Erweiterungen, da Engpässe von Seiten der RAM-Hersteller vorhanden sind, zum zweiten muß der Preis neu kalkuliert werden, so daß es zwangsläufig zu einer Preiserhöhung kommen muß. Kosteten die D-RAMs vor Wochen noch ca. 6.- DM, muß der Kunde zur Zeit den doppelten bis dreifachen Preis bezahlen.
(Hätte man doch bloß vor einem Jahr kiloweise Ram-Chips gebunkert)



PROFEX SE 2000

Eine 2 MByte-Speichererweiterung für den Amiga 500 bietet die Firma Electronic Handel KG an. Die Erweiterung mit der Bezeichnung PROFEX SE 2000 ist in Form und Farbe dem Amiga 500 angeglichen. Sie wird seitlich am A500-Expansionport angeschlossen, ist autokonfigurierend und abschaltbar. Weitere Erweiterungen sind anschließbar, da das Gerät über einen durchgeschleiften Bus verfügt. Im Inneren finden sich 64 D-RAMs mit einer Zugriffszeit von 120 ns. Ein unverbindlicher Verkaufspreis von 1398.- DM wird von der Firma angegeben.

Anbieter:
Electronic Handel KG
Niedernhart 1
8391 Tiefenbach

3.5" Diskettenlaufwerk

Electronic Handel KG bietet des weiteren ein 3.5" Laufwerk für die Amiga-Modelle 500 und 2000 an. Das Super-Slim-Line-Laufwerk mit dem Abmessungen 28.5 x 104 x 202 mm besitzt einen durchgeschleiften Bus, so daß bis zu drei weitere externe Laufwerke an beiden Rechnern betrieben werden können. Das Design des PROFEX DL 1015-Laufwerkes lehnt sich an das von Commodore Amiga 500/2000 an.

Anbieter:
Electronic Handel KG
Niedernhart 1
8391 Tiefenbach

ADRUM

Seit kurzem gibt es eine weitere Schlagzeug-Maschine für den Amiga von der bis dato unbekannten Firma Bullfrog Productions. Das Programm verwendet digitalisierte Drums. Es können selbst gesampelte Sounds verwendet werden, das IFF-Format wird voll unterstützt. Weiterhin ist es MIDI-kompatibel und unterstützt die Stereo-Fähigkeiten des Amiga. Durch das Installieren des Programms auf die Harddisk ist der Einsatz im professionellen Bereich bestens gewährleistet. Eine Bibliothek mit Soundsamples ist bereits vorhanden. Bemerkenswert: Dabei handelt es sich um Samples vom allseits bekannten und weit verbreiteten Yamaha DX-7.

2 MB-RAMBOX von CAS

Von CAS gibt es jetzt eine neue Rambox, die in einer Version für AMIGA 500 und AMIGA 1000 ausgeliefert wird. Die Speichererweiterung wird beim Booten automatisch erkannt (autokonfigurierend), außerdem hat sie einen Dreifach-Umschalter mit den Stellungen AUS/512KB/2MB.
Auf der mitgelieferten Systemdiskette befinden sich zwei neue Befehle, mit denen es möglich ist, die Startadresse und den Speicherbereich festzulegen. Damit kann softwaremäßig auch der Speicher begrenzt werden, falls ein Programm mit den unendlichen Weiten der Rambox nicht zurechtkommt. Zum Kontrollieren der Speichererweiterung gibt es zusätzlich noch ein RAM-Test-Programm, das in 12 Sekunden den Speicher testet und somit zur Sicherheit für den Anwender dient. Der Preis liegt (momentan) bei DM 1148.-.

Anbieter:
CAS
Sprendlinger Landstr. 71
6050 Offenbach/M.
Tel.: 069-84 20 13 / 06104-6 23 76

FESTPLATTEN für alle AMIGA-Modelle

Eine ganze Palette verschiedener Festplatten ist von der Firma Michael Weisgerber geplant. In der preiswertesten Version kommt ein Seagate ST-225 zum Einsatz. Auf Wunsch werden jedoch auch andere Platten eingesetzt, die dem ST-506-Standard entsprechen. Somit können Speicherkapazitäten bis 170 MB realisiert werden.
Die vom Hersteller angegebenen Zeiten liegen im Bereich der C-Ltd-Festplatte, die in einem zurückliegenden Vergleichstest (Ausgabe 11/87) an erster Stelle lag. Man hat jedoch vor, diese Zeiten weiter zu verbessern, wobei auch Laufwerke mit schnelleren Zugriffszeiten eingesetzt werden sollen. Der Preis für die A2000-Version wird bei DM 898.- liegen, die auf Wunsch auch als Filecard geliefert wird. Die A500- bzw. A1000-Version wird DM 998.- kosten, da hier noch ein Gehäuse und ein Netzteil mitgeliefert werden.
Die Preise betreffen hierbei die Grundversion mit einer 20 MB-Seagate-Platte. Der Aufpreis für 30 MB wird bei 150.- DM liegen.

Michael Weisgerber
Hard & Soft
Rathausstraße 2
6551 Fürfeld
Tel.: 06709-778

STUDIO COLLECTION

Von HS&Y gibt es eine Sammlung grafisch anspruchsvoller Vorlagen, Muster und Motive für Bilder und Animationen. Die künstlerische Gestaltung und Zusammenstellung der Motive kommt von Labenski-Design Köln. Vier Disketten, deren Preise zwischen DM 59.- und 79.- liegen, sind bereits fertiggestellt. Die Themen sind:

Vol. 1: Human/Robot-Motion
Vol. 2: Texturen (Strukturen und Muster)
Vol. 3: Bäume
Vol. 4: Mix (Videotitel und Schriften)

Heinrichson Schneider & Young
Herderstraße 94
5000 Köln 41
Tel.: 0221-43 16 87

VIZAWRITE V1.09

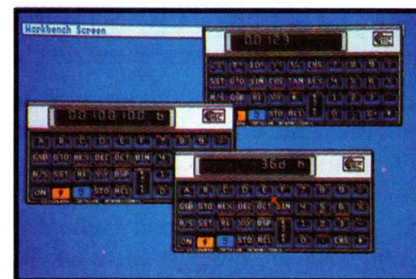
Auch bei VIZAWRITE brachte der Frühling neue Früchte zum Vorschein. Version 1.09 ermöglicht es nun, einem Zeichensatz Steuercodes zuzuweisen, so daß verschiedene Schriftattribute oder Zeichensätze des benutzten Druckers angesprochen werden können. Auf diese Weise ist mit einem Farbdrucker auch ein einfaches Wechseln der Farbe denkbar. Außerdem wurden Fehler behoben und einige Funktionen beschleunigt.
Das Update kann gegen Einsendung der Originaldiskette kostenlos bezogen werden. Zusätzlich gibt es eine Fontdiskette mit 20 Fonts aus der PD-Szene, die ebenfalls gegen Zusendung einer Diskette erhältlich ist.
Schweiz:
MICROTRON
Bahnhofstraße 2
2542 Pieterlen
Deutschland:
DTM
Poststraße 25
6200 Wiesbaden

GFA-BASIC 3.0

Entgegen der Meldung in der letzten Ausgabe wird GFA-Basic doch schon Mitte des Jahres erscheinen. Wir rechnen deshalb damit, bereits in der nächsten oder übernächsten Ausgabe dieses interessante Basic vorstellen zu können. Zur Zeit wird noch an den besonderen AMIGA-Features, dem R.C.T. (Requester Contruction Tool) und auch am Handbuch gearbeitet.

HP-Emulatoren

Zwei neue Programme, AMNIX und DX SERIES, sind von Discovery-Software ins Leben gerufen worden.
Amnix ist eine neue C-Shell für den Amiga. Sie beinhaltet über 40 speicherresidente (33Kb) Kommandos. Zahlreiche weitere Funktionen wie z.B. eine "Command Line History" sind integriert.
DX SERIES ist eine Umsetzung der Hewlett-Packard-Kalkulatoren HP-11 und HP-16. Sie sind nicht nur perfekte Simulationen dieser beiden programmierbaren Taschenrechner, sondern enthalten darüber hinaus Verbesserungen, wie "History Windows", d.h., daß sich der Rechner die gesamten Rechenschritte merkt und bei Bedarf noch einmal ausführt.



AMIGA TOOLS

Von CSJ Computersoft GmbH ist ein Programm mit verschiedenen nützlichen Funktionen erschienen. Interessantester Teil ist sicherlich der VIRUS FINDER, der die momentan bekannten Viren aufstöbert und auf Wunsch den Bootblock neu schreibt.
Der RAM DELETER (NoFastMem) reduziert den RAM-Bereich des AMIGA auf 512KB. Diese Einstellung ist resetfest und gestattet es somit, einige problematische Programme laufen zu lassen, ohne die Speichererweiterung auszubauen. Des weiteren gehören ein schnelles Formatierprogramm und ein Kopierprogramm zum Lieferumfang von AMIGA TOOLS.
Der Preis liegt bei DM 49,95.
CSJ COMPUTERSOFT GmbH
An der Tiefenriede 27
3000 Hannover 1
Tel.: 0511- 88 63 83

2.2 MB im Amiga intern

Bisher war der Amiga 500-Anwender gewohnt, Erweiterungen mit mehr als 512 KByte auf den Expansionport zu stecken. Diese Zeiten sind vorbei. Die 1.8 MByte Erweiterung sitzt fest im Innern des AMIGA. GIGATRON ist einen anderen Weg gegangen und nutzt den Amiga 500-Speichererweiterungsport, der unten im A500-Gehäuse befindet. Die Karte besitzt in etwa die Größe anderer 512 KByte-Erweiterungen, aus diesem Grunde treten keine Probleme mit der Unterbringung der Karte auf, es sei denn, man hat vergessen die alte herauszunehmen.

Hinein damit

Der Einbau ist denkbar einfach: Die untere Abdeckklappe entfernen und die Karte auf den freigelegten Bus stecken - fertig. Dem Anwender stehen jetzt weitere 512 KByte zur Verfügung. Um in den Genuß der ganzen 1.8 MByte zu gelangen, muß der A500 geöffnet, der "GARY"-Chip aus dem Sockel entfernt und die mitgelieferte Adapterplatine eingesteckt werden. Danach muß man den "Gary-Chip" wieder in den freien Sockel der Platine und ein Kabel zwischen Adapterplatine und Speichererweiterung stecken.

Kompakt und sparsam

Der geringe Platzbedarf liegt an der Verwendung von 16 1MBit-Chips, die anstelle der "normalen" 256KBit-RAMs verwendet wurden. Damit der Amiga 500 die kompletten 1.8 MByte auch erkennt, ist eine Steuerlogik vonnöten. Dazu dient ein PAL-Chip, der auf der kleinen Adapterplatine zu finden ist und die Adressierung in die Wege leitet.

Die autokonfigurierende Karte verfügt außerdem über eine akkugepufferte Echzeit-Uhr, die wie üblich über die CLI-Kommandos angesprochen wird und über einen Schalter, mit dem bei Programmen, die nur mit 512 KByte laufen, die Erweiterung einfach abgeschaltet werden kann. Die Uhr bleibt hingegen voll aktiv.

Für den Amiga 1000 wird die Karte in die Sidecar eingebaut.

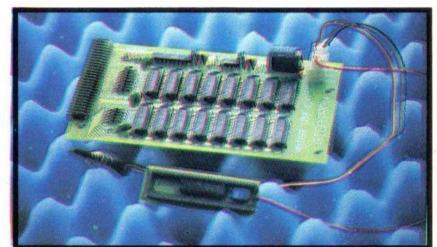
Fazit

Die 1.8 MByte-Speichererweiterung hat einige Vorteile gegenüber ihren Kontrahenten. Der Einbau ist denkbar einfach.

Trotz zusätzlichen 1.8 MByte bleibt der Expansionport frei für anderweitige Erweiterungen, außerdem wird das Gesamtbild des wunderschönen, harmonisch gestylten AMIGA nicht gestört. In der Testphase arbeitete der neue Speicher problemlos, auch an dem Preis konnten wir, bedenkt man, daß teure 1 MBit-RAMs die Platine "schmücken", nichts aussetzen.

Anbieter: GIGATRON

Preis: 1197.- DM (A1000 und A500)

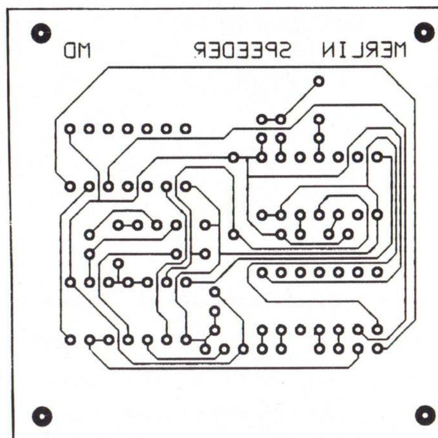


Die Erweiterung findet ihren Bestimmungsort im Speichererweiterungsport des Amiga 500.

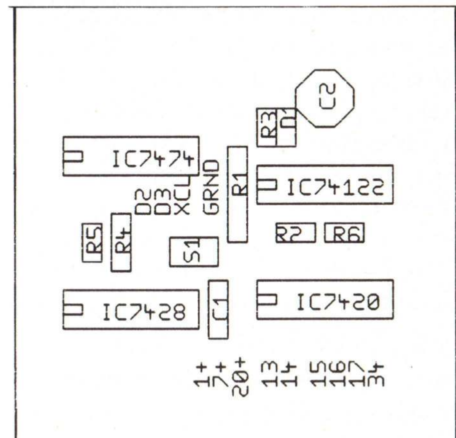
ENDE

Korrektur zum Kickstarter in der letzten Ausgabe

Leider wurde in der letzten Ausgabe eine Vorabversion des Platinenlayouts sowie des Bestückungsplans abgedruckt. Im Schaltbild wurde der Anschluß für die Versorgungsspannung vertauscht (Pin 1 der CIA ist Masse und Pin 20 ist +5V). Bitte entnehmen Sie der hier abgedruckten Berichtigung das entgültige Platinenlayout und den dazugehörigen Bestückungsplan.



Platinenlayout



Bestückungsplan


SupraDrive
Hard Disk
Amiga
20 MB
X-tension
Hard Disk
40MB

FESTE PLATTELN

VON ANDRAS KRÄMER

FESTPLATTEN FÜR DEN AMIGA

Eine Festplatte mit dem von ihr gebotenen Speicherplatz und ihren kurzen Zugriffszeiten ist sicher der Traum vieler floppygeplagter Amiga-Anwender, erleichtert ein solcher Massenspeicher die Arbeit doch sehr und reduziert lange Ladezeiten, die bei intensiver Arbeit mit dem Computer sehr viel Zeit stehlen können. Dies erschien uns Grund genug, in einem Test einige gängige Modelle unter die Lupe zu nehmen.

Der Redaktion stand für diesen Test eine ganze Reihe von Festplatten zur Verfügung. Darunter waren auch Modelle, die interessante Features bieten, so z.B. eine Autoboot-Festplatte mit 40 MByte für den Amiga 1000. Weiterhin testeten wir eine 80 Mbyte-Platte für den 1000er und 65 MByte-Platte für den A500 sowie eine weitere 20 MByte-Platte für den kleinsten in der Amiga-Familie. Jede der Platten mußte zeigen, ob sie ihr Geld wert ist. Alle Geräte wurden auf Herz und Nieren getestet; Alltagstauglichkeit, Lieferumfang, Inbetriebnahme und etliches mehr wurde genau untersucht.

Bisher hatten Festplatten für den Amiga nicht gerade den besten Ruf, hatten sie im Vergleich mit anderen Systemen doch immer das Nachsehen. Ladezeiten der Amiga-Festplatten waren zwar gegenüber der Diskette kürzer, aber immer noch lang gegenüber den Ladezeiten von MS-DOS-Platten. Ein weiteres entscheidendes Manko waren die mangelnden Booteigenschaften, der "Umweg" über die Diskette mußte immer beschritten werden. Längere Wartezeiten waren auch hier vorprogrammiert. Der Amiga-Anwender schaute neidisch auf die MS-DOS-Welt, wo solche Probleme völlig unbekannt sind.

Auch gab es hier und da anderweitige Probleme, z.B. mit dem Anschluß und der Inbetriebnahme der Platten.

X-TENSION, BOOTEN VON DER FESTPLATTE. EINE NEUE GENERATION.

Der Redaktion stand diese Festplatte einer neuen Generation mit zugehöriger Software zum Test zur Verfügung. Das Paket sollte alle bisherigen Kritikpunkte von Amiga-Festplatten vergessen machen. Commodore verspricht zwar mit der Einführung der KICKSTART und Workbench Version 1.4 ein Beheben zumindest der mangelnden Booteigenschaften, aber wieder einmal waren Fremdfirmen schneller. Natürlich gingen wir mit viel Skepsis in den Prüfstand, stellte doch das Angekündigte eine Revolution in der Amiga-Welt dar.

Aus deutschen Landen...

stammt die Entwicklung des Systems. Die Firma Frank Elektronik GmbH aus Nürnberg hat die X-Tension-Harddisk ins Leben gerufen.

Doch zunächst zum Lieferumfang: Der Kunde erhält für sein Geld die eigentliche Festplatte, einen Hostadapter, der auf den Expansionport des Amiga gesteckt wird, etwa 10 Blätter Dokumentation, die zum einen die verschiedenen Bootinstallationen und zum zweiten die Inbetriebnahme des Gerätes erläutern, sowie zwei Disketten, auf denen sich die benötigte Software findet. In unserer Lieferung fanden wir zusätzlich zwei Eproms, die für einen Aufpreis zu haben sind, doch dazu später mehr.

Der erste Eindruck

Die X-Tension Festplatte zeigt sich nicht gerade im Hochzeitsgewand. Wer auf seinem Schreibtisch schon unter akutem Platzmangel leidet, der wird mit der Plazierung der Platte seine Schwierigkeiten haben. Äußerst protzig gibt sich das quadratische, etwa 10 cm hohe Gehäuse, das ganz in Kunststoff gehalten ist. Sind Design und Farbwahl des Festplatten-Gehäuses noch einigermaßen ansprechend, so zeigt sich der Hostadapter dem Betrachter in einen zwar kleinen, aber häßlichen Aluminiumgehäuse.

Ans Werk!

Die Installation ist denkbar einfach: den Hostadatper auf den Expansionport gesteckt, die Verbindung zwischen Adapter und Platte hergestellt, Netzkabel angeschlossen, und fertig ist die Geschichte. Danach nimmt man seine Workbench-Diskette, mit der man normalerweise arbeitet, fertigt eine Kopie an und startet eine Batch-Datei, die sich auf einer der mitgelieferten Disketten befindet. Daraufhin werden alle notwendigen Dateien auf die Workbench-Diskette kopiert.

In der Datei “mountlist” kann man die Definition der Festplatte leicht ändern; so können beispielsweise mehrere Partitionen definiert werden. Unser Testgerät, eine 40 MByte-Platte, haben wir in zwei Partitionen zu 10 und zu 30 MByte aufgeteilt. Auffallend war, daß in der “mountlist” kein Interleave-Faktor (als Interleave-Faktor wird die Blockstruktur eines Tracks bezeichnet; bei einem Interleave-Faktor von 0 sind alle Blöcke aneinander gereiht: 012..., bei einem Faktor von 1: 0213...) zu finden war. Der logische Schluß ist ein Interleave-Faktor von 0, was auf maximale Geschwindigkeit hoffen ließ.

Nach “mount dh0:” bzw. “mount dh1:” ist die Platte angemeldet und kann formatiert werden. Der Format-Befehl wird zu diesem Zweck herangezogen. Ist die Formatierung abgeschlossen, kann die X-Tension mit eigenen Dateien wie gewohnt bespielt werden. Beispielsweise kann man die Workbench-Diskette auf die Platte kopieren, die Startup-Sequence in der Boot-Workbench ein wenig modifizieren, und schon steht dem Anwender die Festplatte als System-Diskette zur Verfügung. Doch all das ist noch nichts Neues, denn das können andere Platten auch.

Booten von Festplatte

Die Dokumentation beschreibt zwei Möglichkeiten hierfür, jedoch ist das Booten zunächst nur für den Amiga 1000 von Interesse. A500 und A2000

Bild 1: Die 40 MByte X-Tension mit Hostadapter.



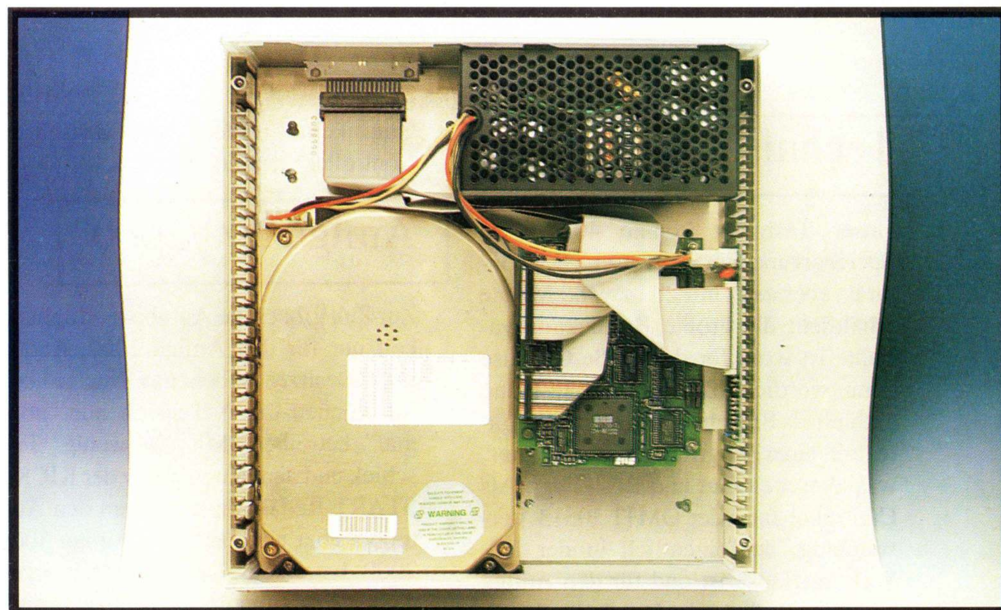
haben hier das Nachsehen, da sich das Betriebssystem in ROMs befindet, für den Bootvorgang aber die KICKSTART auf Diskette vorhanden sein muß.
Zum einen kann die X-Tension dazu veranlaßt werden, nach dem Booten der KICKSTART direkt mit dem Laden von Festplatte zu beginnen, zum zweiten kann der globale Bootvorgang, also das Booten von KICKSTART und Workbench, von Festplatte erfolgen. Doch wie geht das vonstatten? Zum ersten Weg: Hierzu muß man zunächst, wie gewohnt, das System booten. Dabei verwendet man eine übliche KICKSTART-Diskette und die Workbench, welche die Festplatte anspricht. Auf der mitgelieferten Diskette muß man danach das Programm XStall mit einer bestimmten Option starten, woraufhin man aufgefordert wird, eine formatierte Diskette in Laufwerk df0: zu legen. Auf diese Diskette wird eine modifizierte KICKSTART geschrieben, die ab sofort das Laden der Workbench von Platte erlaubt, vorausgesetzt, man bootet das System mit der "neuen KICKSTART". Auf der Festplatte sollten natürlich alle notwendigen Dateien vorhanden sein.

Autoboot-Programme

Natürlich stellt sich hierbei die Frage: Wie verhält es sich mit Autoboot-Programmen, die normalerweise statt der Workbench eingelegt werden? Um auch solche Programme verwenden zu können, muß man nach einem RESET lediglich die rechte Maustaste drücken. Daraufhin wird nicht von Platte geladen, sondern der Rechner verlangt wie gewohnt die Workbench in df0:.

Booten ohne Diskette!

Des weiteren ist es möglich, komplett von der X-Tension zu booten (KICKSTART und Workbench). Hierzu ist jedoch ein Eingriff in den Amiga 1000 vonnöten, denn die zuvor schon erwähnten Eproms kommen zum Einsatz. Sie müssen anstelle der Boot-ROMs in den Rechner eingesetzt werden, was natürlich einen Garantieverlust zur Folge haben kann. Aus diesem Grunde sollte man den Austausch von einem Fachmann durchführen lassen. Bevor man den Austausch vornimmt, sollte man genauso vorgehen wie oben beschrieben, jedoch verlangt das Programm XStall hierbei eine andere Option. Die momentane KICKSTART wird jetzt nicht mehr auf eine Diskette gesichert, sondern auf der X-Tension verewigt.
Nach Einschalten der Platte und des Amiga muß jetzt keine Diskette mehr eingelegt werden, das System wird komplett vom Speicherriesen gebootet. Es ist schon ein komisches Gefühl, keine Diskette mehr einlegen zu müssen. Will man ein Autoboot-Programm laden, so muß man genauso wie oben beschrieben die linke Maustaste drükken, und der Rechner verlangt wie gewohnt die Workbench-Diskette in Laufwerk df0:. Mancher Amiga 1000-Besitzer wird vielleicht die Hand vermissen, die die verlangte Diskette hält, denn sie ist einem kurzen Kommentar gewichen. Möchte der Anwender eine andere KICKSTART-Version laden, muß nach Einschalten des Rechners wiederum die linke Maustaste gedrückt gehalten werden, und man kann wie gewohnt das Betriebssystem von Diskette laden.


Bild 2: Das Innenleben der 40-MByte X-Tension ist aufgeräumt, besitzt aber viel Freiraum.

Geschwindigkeit

Frank Elektronik bezeichnet die X-Tension unter anderen auch als "Fast-Disk". Die mutige Namensgebung mußte aber erst bestätigt werden. Gleich vorweg: Die Firma war nicht zu dreist, die X-Tension glänzt mit äußerst kurzen Ladezeiten. Eine kleine Tabelle gibt Auskunft über die Geschwindigkeit der Platte; zum Messen wurde die Ladezeit des AmigaBASIC herangezogen. Auf sogenannte Benchmarks haben wir absichtlich verzichtet, denn viele Anwender können mit Transfer- oder Stepraten wenig anfangen; ein Praxisbeispiel ist aus diesem Grunde naheliegend. Das knapp über 103 KByte lange Programm wurde von RAM auf Platte und von der Platte ins RAM kopiert sowie geladen (näheres siehe Tabelle 1).

Schneller mit dem "KICKSTARTER"

In Ausgabe 5/88 der KICKSTART hatten wir eine kleine Hardwareschaltung vorgestellt, die den Systemtakt des Amiga heraufsetzt. Mit diesem Speeder erreichte die Platte eine zusätzliche Geschwindigkeitssteigerung, die bei etwa 20 Prozent lag. Das einzige Problem hierbei war, daß der Amiga nach einem Reset "hängenblieb" und das erneute Booten verweigerte. Dies liegt wohl an Timing-Problemen zwischen dem Festplatten-Controller und dem 68000er. Der Amiga muß ausgeschaltet werden, was nicht sonderlich ins Gewicht fällt, da lediglich 6-7 Sekunden verstreichen, bis nach der Workbench verlangt wird - vorausgesetzt, die Platte ist fähig, das KICKSTART zu booten.

Technisches

Unser Testgerät konnte 40 MByte Speicherkapazität aufweisen. Das Gerät ist aber auch mit 20 bzw. 80 MByte erhältlich; die größte Bestückungsvariante ist wohl für das große Gehäuse verantwortlich. Die 40 MByte-Platte enthielt als Speichereinheit eine Seagate mit einer Zugriffszeit von 40 ms, was zwischen Gut und Böse liegt. Als Controller fand der OMTI 20513 Verwendung, der eigentlich in der PC-Welt zu Hause ist und für den Amiga "mißbraucht" wurde. Durch die Schaltung des Controllers direkt auf den Amiga-Prozessor-Bus ist unter anderen die recht hohe Geschwindigkeit zu erklären. Durch die Einsparung eines Lüfters ist die X-Tension zwar Leise, wird aber dementsprechend auch sehr heiß. Laut Hersteller sind jedoch keine Probleme aus diesem Grund zu erwarten, was bei unserem Testgerät auch nicht vorkam.

Nachteile?

Leider besitzt das Paket auch einige Nachteile. Neben dem erwähnten großen Platzbedarf und der mangelden Dokumentation ist der Expansion-Bus des Hostadapters nicht durchgeführt. Der Adapter muß folglich als letzte Erweiterung eingesteckt werden. Besitzt man beispielsweise eine externe Speichererweiterung, die keinen durchgeführten Bus hat, so muß man entweder auf die Platte verzichten oder auf die Erweiterung. Erweiterungen mit durchgeschleiften Bus sind aber auch nicht das Gelbe vom Ei. Zum Test standen uns mehrere Golem 2-MByte-Erweiterungen zur Verfügung. An einigen versagte die Platte ihren Dienst, an anderen wiederum nicht; probieren lautet hier die Devise. Leider läßt sich nichts Eindeutiges sagen. Auf jeden Fall sollte man sich vergewissern, ob Probleme auftreten können. Die Firma Kupke (Golem-Box-Hersteller) tauschte beispielsweise nicht mit anderen Erweiterungen funktionierende Boxen um.

Ein weiterer Nachteil besteht in der Tatsache, daß keine weiteren Festplatten mit dem Controller betrieben werden können. Ist die Platte einmal voll, kann keine weitere über denselben Controller installiert werden.

Amiga 500/2000?

Zur Zeit gibt es die Autoboot-Möglichkeit nur für den Amiga 1000, A500/2000-Besitzer haben das Nachsehen. Sie können die X-Tension nur "normal" betreiben. Mit ein wenig Geschick und dem Know-How der KICKSTART-Redaktion kann man den Autoboot aber auch auf dem Amiga 500/2000 betreiben. Dazu benötigt man einen Eprombrenner, beispielsweise unseren JUNIOR-PROMMER (vorgestellt in KICKSTART 5/88 und 6/88), um die modifizierte KICKSTART auf Eproms brennen zu können, desweiteren noch eine kleine Platine, die die gebrannten KICKSTART-Eproms aufnimmt. Die Platine wird dann anstelle des KICKSTART-ROMs im A500/2000 eingesetzt, und fertig ist die Angelegenheit (das betreffende Platinenlayout wird in einer der nächsten KICKSTART-Ausgaben veröffentlicht).

Für den A500 bietet die Herstellerfirma einen Busadapter an, so daß mit dem Anschluß keine Schwierigkeiten zu erwarten sind. Beim A2000 sieht die ganze Sache weniger rosig aus, eine Anschlußmöglichkeit ist zur Zeit nicht erhältlich.

Fazit

Die X-Tension Festplatte ermöglicht das, was auf anderen Systemen schon lange gang und gebe ist: Das komplette Booten von Festplatte. Commodore verspricht mit der Betriebssystem-Version 1.4 endlich die Einführung dieser Möglichkeit, aber bis zur Auslieferung wird noch einige Zeit vergehen.

Der kleine Schreibtisch wird noch kleiner durch die Abmessungen der X-Tension, so daß sich mancher Amiga-Besitzer ein "Platzbeschaffungsprogramm" zulegen muß. Äußerst erfreulich war die Geschwindigkeit der Platte, läßt sie doch viele Kontrahenten hinter sich. Auch der Preis ist akzeptabel (im Vergleich zu anderen Amiga-Festplatten, wohlbemerkt); mit 1998.- DM schlägt die 40 MByte-X-Tension zu Buche, für die 20 MByte-Platte werden 1498.- DM verlangt und für die 80 MByte-Version 3498.- DM. Die zwei Boot-Eproms kosten zusätzliche 48.- DM. Ein Amiga 500-Besitzer muß auf die Eproms verzichten, benötigt aber 48.- DM für den Busadapter.

Amiga 1000-Besitzern kann zur X-Tension geraten werden. Die Platte arbeitete die komplette Testphase ohne Probleme, lediglich der Hostadapter ohne durchgeschleiften Bus stellt einen Wermutstropfen dar.

Tabelle 1: Die Geschwindigkeit wurde anhand der Ladezeit des AmigaBasics festgestellt.

	X-tension	mit Speeder	Diskette	RAM
AmigaBASIC Festplatte -> RAM	2.60	2.11	10.97	--
AmigaBASIC RAM -> Festplatte	4.40	3.62	25.21	--
AmigaBASIC ->laden	2.51	2.05	10.15	1.66

Alle Zeitmessungen in Sekunden

X-Tension-Festplatte HD1040

+ schneller Datenzugriff
+ komplettes Booten von Festplatte möglich
+ andere KICKSTART(Workbench-) Versionen ebenfalls ladbar
+ leise
+ solide Verarbeitung
+ leichte Installation der Autoboot-Routinen

- großes Gehäuse
- keine weiteren Festplatten anschließbar
- schlechte Führung des Hostadapters am Expansionport
- mangelnde Dokumentation

Anbieter:
Frank Elektronik GmbH
Matthiasstr. 3
8500 Nürnberg 80 .Tel. 0911-327732

Preis: 20 MByte 1498.- DM
40 MByte 1998.- DM,
80 MByte 3598.- DM
Booteproms 48.- DM
A500 Busadapter 48.- DM

DER PHOENIX DER FEST-PLATTEN ?

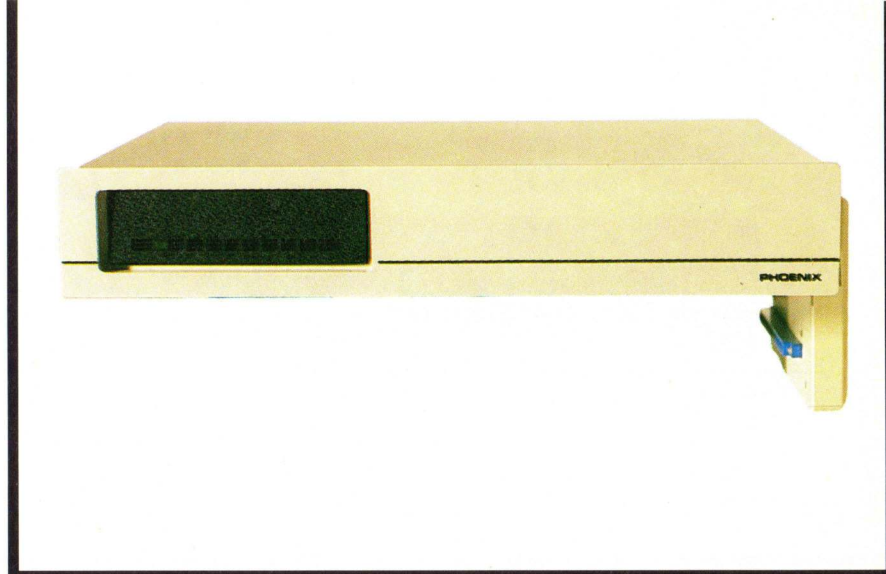


Bild 3: Die Amiga 1000-Festplatte von Phoenix muß auf den Rechner gestellt werden.

Aus Amerika stammt eine weitere Festplatten-Serie, deren Modelle in verschiedenen Kapazitäten erhältlich sind. Der Redaktion stand eine 80 MByte-Platte für den Amiga 1000 und eine 65 MByte-Platte für den Amiga 500 zum Test zur Verfügung.

Der Lieferumfang

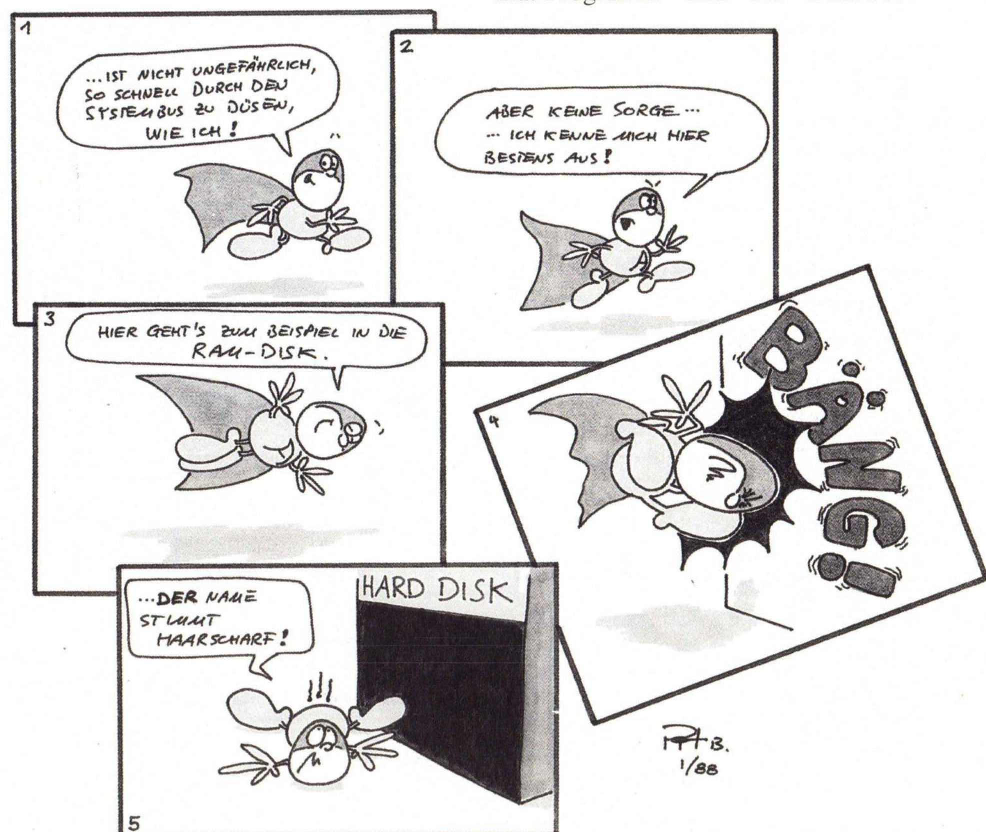
Bei diesen Festplatten ist man zunächst über die Größe der Kartons erstaunt. Beim Auspacken kommen dann auch zwei recht große Geräte zum Vorschein. Die Amiga 1000-Festplatte besitzt in etwa die Größe des Rechnergehäuses, die 500er-Platte ist etwas kleiner gehalten, hat aber dennoch stattliche Ausmaße. Der Sinn des großen Gehäuses der 1000er-Platte besteht darin, daß die Platte einfach auf dem Rechner plaziert und der Monitor wie gewohnt darauf gestellt wird. Der Vorteil dieser Methode liegt auf der Hand: Die Platte nimmt keinen wertvollen Platz auf dem (oftmals) schon stark unter Platzmangel leidenden Schreibtisch ein. Bei der Amiga 500-Platte muß jedoch ein passendes Plätzchen auf dem Schreibtisch gefunden werden. Im Karton finden sich des weiteren ein Hostadapter mit Kabel, eine Boot-Diskette und ein englischsprachiges Handbuch. Die Hostadapter finden ihren Bestimmungsort an den Expansionports der Rechner. Über ein Kabel werden sie dann mit den Platten verbunden.

Ans Werk

Der Anschluß beider Platten bereitete keine Probleme, so daß die Arbeit mit den Platten gleich beginnen konnte: Festplatte an, Rechner und Monitor an und statt der Workbench-Diskette die mitgelieferte Phoenix-Diskette eingelegt (beim Amiga 1000 zunächst die KICKSTART), woraufhin die Platte installiert wird und sich dann mit dem Disketten-Icon meldet. Die 80 MByte-Platte hatte vier, die 65 MByte-Patte drei Partitionen. Die Boot-Diskette weist der Festplatte automatisch den logischen Gerätenamen "sys:" zu und definiert auch gleich die entsprechenden Suchpfade zur Festplatte. Danach kann sie getrost aus dem Laufwerk df0: entfernt werden.

Wie eine solche Installation vorgenommen wird, wird im mitgelieferten Handbuch genau beschrieben, so daß man eigene Vorstellungen (beispielsweise das Anlegen anderer Partitionen) leicht verwirklichen kann.

Läßt man sich den Inhalt der Festplatten ausgeben, treten noch weitere interessante Programme zutage, die in der Dokumentation gänzlich unerwähnt bleiben. Neben einigen PD-Programmen, hauptsächlich Demos, findet der Käufer ein SCSIFormat-Programm, das uns schon von den Festplatten der Firma CLtd bekannt vorkam. Nach Starten bestätigte sich diese Vermutung. Mit dem Programm kann man die Festplatte formatieren, wozu jedoch einige Angaben eingegeben werden müssen, die man in der Mountlist-Datei im Ordner "devs" einträgt. Des weiteren befinden sich noch ein Fast-Format-Programm und ein Park-Pro-



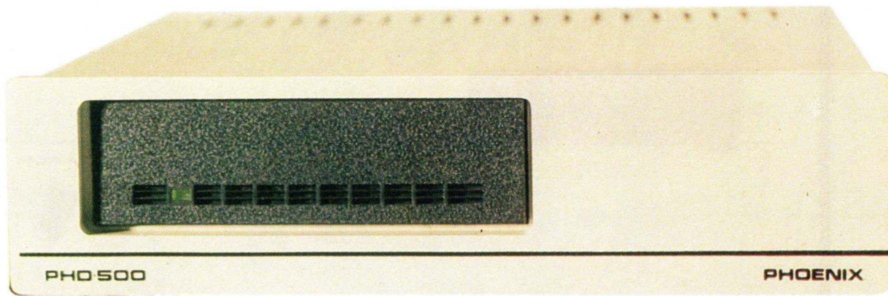


Bild 4: Die Amiga 500 Festplatte ist solide verarbeitet.

gramm, das die Platte bei einem Transport parkt und somit vor Beschädigungen schützt, auf den Platten.

Verarbeitung

Beide Festplatten sind in einem Metallgehäuse untergebracht und sehr solide verarbeitet. Ein Manko stellt bei der Amiga 1000-Platte der Hostadapter dar, der nicht ganz in das Bild der Platte passen möchte. Auch der Anschluß machte gewisse Probleme, die Platte versagte zunächst ihren Dienst. Der Fehler war im mangelhaften Kontakt der Expansionport-Buchsen zu finden. Ein erneutes starkes Drücken und Schieben stellte den erwünschten Kontakt jedoch her, und die Platte versah ihren Dienst fortan ohne Probleme. Mit der 500er-Platte traten beim Anschluß keine Probleme auf.

Geschwindigkeit

Beide Festplatten besitzen ein Seagate-Laufwerk mit 40 ms Zugriffszeit. Allerdings läßt die Zugriffszeit der Platten keine Schlüsse auf die eigentliche Geschwindigkeit zu, denn diese hängt von der Datenübertragungsrate ab. Wie bei allen in diesem Heft getesteten Platten verzichten wir auf Angaben der Stepgeschwindigkeit oder ähnliches, da wir die Datenübertragungsgeschwindigkeit anhand eines praktischen Beispiels ermittelten. Die Geschwindigkeit der 80 MByte-Platte kommt nicht an die der X-Tension heran. Die Platte für den Amiga 500 ist noch etwas langsamer als die für den A1000, was wohl daran liegt, daß ein anderer Controller für den A500 verwendet wurde. Nähere Informationen sind der Tabelle 2 zu entnehmen.

Sieben weitere Platten

Laut Hersteller können bei der Amiga 1000-Version bis zu sieben weitere Platten über denselben Controller betrieben werden. Beim Amiga 500 ist der Anschluß weiterer Platten nicht vorgesehen.
Beide Hostadapter besitzen einen durchgeschleiften Expansionbus. Die Funktion des Busses ist beim A1000 eingeschränkt; einige Erweiterungen versagten ihren Dienst, andere wiederum ließen keine Beeinträchtigungen erkennen. Der gleiche Efekt war beim A500 festzustellen - probieren lautet auch hier die Devise.

Fazit

Beide Platten verrichteten ihren Dienst ohne Schwierigkeiten, im alltäglichen Einsatz gibt es also keine Probleme. Die Plazierung der A1000-Platte ist durchdacht und raubt keinen weiteren Platz auf dem Schreibtisch, der A500-Platte muß jedoch viel Platz eingeräumt werden. Die Installation ist problemlos und einfach.
Die Lüftergeräusche der A1000-Phoenix sind gewöhnungsbedürftig, das Laufgeräusch der Platte hingegen ist gering. Lediglich beim Ein- bzw. Ausschalten der Platte vernimmt man etwas lautere Geräusche. Dies gilt auch für die 500er-Platte, bei der es allerdings keine Lüftergeräusche gibt, da kein Lüfter vorhanden ist.
Die Dokumentation ist knapp und in englischer Sprache; hier müßte Abhilfe geschaffen werden.
Auf einen Nenner gebracht: Die Phoenix-Festplatten sind zuverlässig und solide in der Verarbeitung. Sie bieten nichts neues, beschleuigen aber die tägliche Amiga-Arbeit.
Nachtrag: Demnächst werden die A500-Platten in Super-Slim-Line-Gehäusen fertiggestellt sein.

Phoenix-Festplatten

+ praktische Plazierung der A1000-Platte
+ solide verarbeitet
+ bis zu sieben Platten beim A1000 zusätzlich anschließbar
+ Expansionbus durchgeführt
+ leise A500-Platte

- großes Gehäuse der A500-Platte
- laute Lüftergeräusche der A1000-Platte
- mangelnde Dokumentation in Eng lisch

Anbieter:
IM
Borsigallee 18
6000 Frankfurt 60
Tel: 069-410071/72

Preis: 3495.- DM (80 MByte)
2495.- DM (65 MByte)
1895.- DM (40 MByte)
1495.- DM (20 MByte)

Alle Platten sind in einer A1000- und einer A500-Version erhältlich.

Tabelle 2: Der Geschwindigkeitsvergleich der Phoenix-Platten

	Phoenix A1000	mit Speeder	Phoenix A500	Diskette	RAM
AmigaBASIC Festplatte -> RAM	5.06	4.45	6.86	10.97	--
AmigaBASIC RAM -> Festplatte	8.14	7.13	12.23	25.21	--
AmigaBASIC ->laden	4.91	4.05	5.72	10.15	1.66

Alle Zeitmessungen in Sekunden

SUPRADRIVE, 20 MBYTE AM AMIGA 500

Für den Amiga 500 stand uns eine weitere Festplatte zum Test zur Verfügung, das SupraDrive. Es besitzt eine Speicherkapazität von 20 MByte und gehört damit zu den kleineren Speicherriesen.

Der Lieferumfang

Im Lieferumfang befinden sich neben der Festplatte der Hostadapter, der auch als SCSI-Interface bezeichet wird, ein 60 Seiten starkes DinA5-Handbuch und eine Diskette, die einige Überraschungen zutage brachte.
Der Anschluß macht bei der SupraDrive keine Probleme; den Hostadapter an den Expansionport des A500 gesteckt, das Netzkabel in die Steckdose - fertig.

Für Anfänger geeignet

Bevor der Rechner eingeschaltet wird, muß zunächst die Platte mit Strom versorgt, sprich eingeschaltet werden. Statt der Workbench-Diskette wird die mitgelieferte SupraDrive-Diskette eingelegt, woraufhin das System geladen, die Platte angemeldet und als logisches Gerät “sys:” deklariert wird. Der Anwender erblickt nach einer Weile die Workbench mit vier Platten-Piktogrammen, woraus sich leicht schließen läßt, daß die Platte in vier Partitionen unterteilt ist, was bei einer 20 MByte-Platte natürlich Ansichtssache ist - 5 MByte pro Partition? Der ersten Partition wird der logische Gerätename “sys:” zugewiesen, so daß alle weiteren Systemzugriffe auf der Platte getätigt werden.
Zuerst schauten wir uns die Boot-Diskette näher an. Die Ausgabe des Verzeichnisses brachte eine Überraschung: Auf der Diskette befindet sich das bekannte Programm CLImate, das das CLI in eine grafikorientierte Oberfläche verwandelt. Dieses Programm ist ein Bonus der SupraDrive. Dem CLImate ist ein Extra-Kapitel im Handbuch gewidmet, so daß Fragen zur Bedienung schnell beantwortet sind. Weiterhin findet der Käufer auf der Diskette zwei Programme, die auf die Namen ClockSet und ReadClock getauft sind. Die Programme unterstützen die batteriegepufferte Uhr im Hostadapter. Mit Clockset kann die Uhr gestellt und mit ReadClock die Zeit an das Amiga-Betriebssystem übergeben werden. Leider ließ sich die Uhr nicht einstellen; wurde das Programm ReadClock von der Workbench aus gestartet, verabschiedete sich der Rechner mit einem Software Failure und einem nachfolgenden Guru. Der Grund der Funktionsstörung war schnell gefunden: Die Uhr, oder besser das Uhrenmodul, war im Hostadapter unauffindbar.
Weiterhin befindet sich auf der mitgelieferten Diskette ein Programm mit dem Namen Supramount. Dieses ist zuständig für das “Mounten” der Platte und das Aufteilen der vier Partitionen in 2, 2 * 5 und 8 MBYte. Wer die Platte in andere Partitionen organisieren möchte, muß den normalen Weg über die Mountlist-Datei gehen oder ein weiteres Programm aufrufen, das im Ordner “System” zu finden ist und mit dem die neuen Parameter der Platte eingestellt werden können. Das letzte nützliche Programm ist ein Festplatten-Parkprogramm, welches vor einem Transport aufgerufen werden sollte, damit Beschädigungen der Platte vorgebeugt wird.



Bild 5: Die Abmessung der Supra-Drive.

Verarbeitung

Die SupraDrive-Festplatte ist in einem soliden Metallgehäuse untergebracht. Dies gilt auch für den Hostadapter, der in der Form dem A500-Gehäuse angeglichen ist. Leider wirkt der Adapter etwas protzig und besitzt in etwa die Ausmaße der Platte selbst.

Geschwindigkeit

Festplatten sind natürlich generell schneller als Disketten-Laufwerke, aber enorme Geschwindigkeitsunterschiede bestehen auch hier. Dies liegt zum einen an den verschiedenen Zugriffszeiten der Platten sowie vor allem an der Verarbeitung der Daten im Controller und im Hostadapter. Die SCSI-Platte liegt mit ihrer Geschwindigkeit im Mittelfeld, selbstverständlich in Bezug auf Amiga-Platten. Näheres läßt sich in Tabelle 3 nachlesen; auch hier wurde das AmigaBasic mit knapp über 100 KByte zur Zeitnahme herangezogen. Mit einer Erhöhung des Taktes mit unserem KICKSTARTER wird auch diese Platte um etwa 20 % schneller.

Fazit

Die SupraDrive-Festplatte gab im Test keinen Anlaß zu Klagen, sie verrichtete ihre Arbeit ohne jegliche Probleme. Der Anschluß ist einfach und problemlos und auch für den Laien keine große Hürde. Lediglich bei einer anderen Aufteilung der Platte (Partitionen) könnte ein Anfänger in Schwierigkeiten geraten. AutoBoot von Festplatte ist unter KICKSTART und Workbench 1.2 nur eingeschränkt möglich, die ersten Zugriffe müssen immer von Diskette stattfinden. Unter der Betriebssystem-Version 1.4 soll auch ein komplettes Booten von Platte möglich sein.

Der Lärmpegel der Platte hält sich in Grenzen, wozu der fehlende Lüfter beiträgt; lediglich beim Ein- bzw. Ausschalten steigt der Lärmpegel ein wenig an.

Die Dokumentation ist leider in englischer Sprache gefaßt, dafür aber ausführlich; sie beschreibt das Arbeiten und den Umgang mit der Platte gut.

Auch am SupraDrive können bis zu sieben weitere Platten über den Kontroller betrieben werden, ohne daß ein weiterer Hostadapter vonnöten ist.

Der durchgeschleifte Expansionport läßt weitere Erweiterungen zu, doch leider heißt auch hier die Devise "probieren". Einige Speichererweiterungen, die zusätzlich zum SupraDrive angeschlossen waren, zeigten zwar den zusätzlichen Speicher an; wurde dieser Speicher jedoch angesprochen, so stürzte der Rechner ab oder brachte die lapidare Meldung "Kein freier Speicher".

Als Bonus ist das Programm CLImate zu nennen, das im Lieferumfang inbegriffen ist, denn der Käufer erhält damit ein leistungsfähiges Utility-Programm.

So bleibt zu sagen: Wem die Diskette zu klein und/oder zu langsam geworden ist, der ist mit der SupraDrive gut bedient.

SupraDrive 20 MByte

+ solide verarbeitet
+ weitere Festplatten anschließbar
+ einfache Inbetriebnahme
+ CLImate im Lieferumfang inbegriffen
+ Expansionport durchgeführt

- englisches Handbuch
- Platte nicht autoparkend
- großer Hostadapter

Anbieter:
PDC
Louisenstr. 115
6380 Bad Homburg
Tel 06172-24748

Tabelle 3: Die Geschwindigkeit der Supra-Drive.

	SupraDrive	*Diskette*	*RAM*
AmigaBASIC Festplatte -> RAM	4.53	10.97	--
AmigaBASIC RAM -> Festplatte	6.02	25.21	--
AmigaBASIC -> laden	4.32	10.15	1.66

Alle Zeitmessungen in Sekunden

ENDE

KICK START

KICK START

KICK START

KICK START

KICKSTART-GRUẞKARTE

Bitte frei-machen



DIE FACHZEITSCHRIFT FÜR DEN AMIGA-ANWENDER

KICKSTART-GRUẞKARTE

Bitte frei-machen



DIE FACHZEITSCHRIFT FÜR DEN AMIGA-ANWENDER

'TOP 12'

Mein Lieblingsspiel

Wettbewerbsbedingungen siehe TOP 12 in diesem Heft

Bitte frei-machen

'MERLIN'- Computer GmbH
Redaktion KICKSTART
'TOP 12'
Industriestraße 26

D-6236 Eschborn

KICKSTART-GRUẞKARTE

Bitte frei-machen



DIE FACHZEITSCHRIFT FÜR DEN AMIGA-ANWENDER

AZTEC-C FÜR AMIGA
VERSION 3.6

Wußten Sie, daß eines der verbreitesten und komplexesten Betriebssysteme - UNIX - in C geschrieben ist ?

NEU! V.3.6

Wußten Sie, daß auch das Betriebssystem des AMIGA größtenteils in C geschrieben wurde ?

Wußten Sie, daß C eine der wichtigsten und modernsten Programmiersprachen ist ?

Wußten Sie, daß Aztec-C einer der schnellsten und leistungsfähigsten Compiler für den AMIGA ist ?

Wußten Sie, daß jetzt Aztec-C in der Version 3.6 verfügbar ist ?

Möchten Sie mehr darüber wissen? Dann schicken Sie uns einen ausreichend frankierten Rückumschlag und Sie erhalten ausführliche Information.

Up-Date-Service für alle MANX-Kunden auch bei uns. Fragen Sie nach!

HIERMIT BESTELLE ICH:

- ☐ AZTEC-C68K/AM-P PROFESSIONAL SYSTEM FÜR DM 398.-
- ☐ AZTEC-C68K/AM-D DEVELOPER SYSTEM FÜR DM 598.-
- ☐ AZTEC-SDB SOURCE LEVEL DEBUGGER FÜR DM 149.-

Versandkosten: Inland DM 7,50 Ausland DM 10,-
Auslandbestellungen nur gegen Vorauskasse
Nachnahmegebühr DM 3,70

- ☐ Vorauskasse
- ☐ Nachnahme

NAME: ____________________

VORNAME: ____________________

STRAßE: ____________________

ORT: ____________________

UNTERSCHRIFT: ____________________

AZTEC-C IST EIN EINGETRAGENES WARENZEICHEN VON MANX SOFTWARE SYSTEMS



MERLIN COMPUTER GMBH
INDUSTRIESTRAßE 26
6236 ESCHBORN
TEL. 06196/481811

VON GERY GLENDOWN

VERSION 1.3

Die neue KICKSTART und WORKBENCH

nämlich die Scheibe, die die neue Kickstart und Workbench V.1.3 / 34.5 enthält. Zwar nicht in dem Maße, wie es die Gerüchteküche verbrach, und offiziell nicht von Commodore zu erhalten, doch mit einigen guten und interessanten Veränderungen. Was ist dran an der "1.3er"? Die Kickstart-Redaktion untersuchte sie für Sie! Hier sind die Ergebnisse:

Ein Gespräch mit Andy Finkel, einem der Betriebssystem-Programmierer bei Commodore, brachte es an den Tag: Eine Assembler-Kickstart ist in absehbarer Zeit nicht zu erwarten (zumindestens nicht von der USA-Abteilung!?). Zwar wurden und werden einige kleinere Routinen in Assembler geschrieben, doch scheuen die Commodore-Leute eine vollständige Umsetzung, da dann Veränderungen für nachfolgende Amigas (!?) zu kompliziert wären. Damit kann man also das Kapitel "Assembler-Kickstart" vorerst abschließen. Doch wie steht es mit den aktuellen Änderungen in Kickstart und Workbench?

Vorweg erst einmal folgende Anmerkung: Auf der von uns vorliegenden Workbench waren eine Reihe kleinere PD-Programme enthalten, von denen Commodore noch nicht weiß, ob diese (rechtlich gesehen) auf der Disk enthalten sein dürfen (wie gesagt, Kick & WB 1.3 sind keine offiziellen Releases!). Doch nun zu den bisher enthaltenen Veränderungen.

Das C-Directory

Auf den ersten Blick fallen hier kaum neue Befehle auf, lediglich "FF", "GetEnv", "GetEnv", "Lock" und "SetAlert" sind total neu. Dabei steht "FF" für "FastFonts", mit dem die gesamte Textausgabe im CLI beschleunigt wird und mit dem man auch andere Fonts verwenden kann (irgendwie kommt mir das bekannt vor, auf den Fish-Disks war's aber nicht...). "SetEnv[ironment]" ist für CLI-Variablen zuständig, so kann man z.B. mit

```
SetEnv DEBUG TRUE
```

der Variablen DEBUG den Text durch True zuweisen und dies dann mittels "GetEnv" in Batch-Files abfragen etc. "Lock" kann einzelne HardDisk-Partitions mit einem Password "verschließen".

Mit "SetAlert" installiert man ein Programm, das die "Recoverable Alerts" bei Layer-Fehlern wirklich zum Zurückkehren bewegt, so daß Fehler in diesem Bereich nicht mehr zum Absturz führen. Des weiteren installiert er einen 68020/68881-Patch.

So weit, so gut, doch bei so vielen Befehlen im C-Directory müssen doch noch weitere Änderungen vorhanden sein. Also mal eben "list" aufgerufen...oh! Ein Programmierfehler - oder...? Da erscheinen auf einmal statt vier acht File-Flags (normal RWED). Dem mußten wir natürlich nachgehen. Dabei kam folgendes heraus:

Die vier neuen Flags (HSPA) haben folgende Bedeutung (wenn auch teilweise noch keine Funktion...):

- H Das Hide-Flag (nein, nicht Jekyll und Hyde...) sorgt dafür, daß ein File beim "Dir"-Befehl nicht sichtbar ist.

- S Das Script-Flag sorgt dafür, daß man von der neuen Shell (ebenfalls auf dieser Disk) Batchfiles ohne "Execute" ausführen kann, also statt "Execute shell-startup" nur noch "shell-startup" schreibt. Dieses Flag war bereits in Funktion.

- P Das Pure-Flag in gesetzter Form war die Voraussetzung dafür, daß man einen Befehl mittels "Resident" (ebenfalls neu) für die Shell einladen kann. Wir setzen diese Flags bei einigen anderen Files, die man dann ohne Probleme ebenfalls mit "Resident" laden konnte. Diese Flag scheint ebenfalls noch nicht implementiert zu sein, genau wie das letzte Flag:

- A Das Archiv-Flag wird für Hard-Disk-Backups verwendet. Beim Anfertigen eines Backups wird dieses Flag gesetzt. Sobald dann ein Schreibzugriff auf ein File stattfindet, wird dieses Flag gelöscht, so daß bei einem neuen Backup nur die wirklich neuen Files gesichert werden müssen.

Im "C"-Directory wurde jedoch noch einiges mehr getan, nämlich die Erweiterung einiger "normaler" Befehle.

Dem aktuellen "Virus-Trend" folgend, hat auch Commodore sich angeschlossen und dem "Install"-Befehl den Modus "Check" hinzugefügt, mit dem man den Bootblock einer Diskette überprüfen kann. Des weiteren kann man mit "NOBOOT" die Diskette ohne BootBlock installieren.

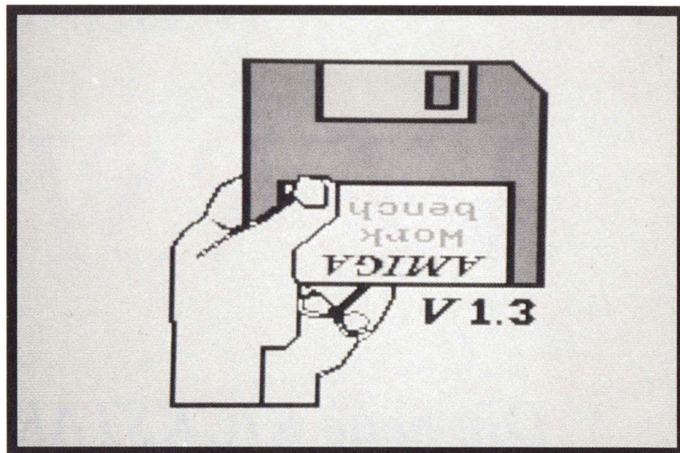
Auch "DiskDoctor" hat einige kleinere Erweiterungen erfahren, was zu einer Erhöhung der Versionsnummer auf "1.3.1" führte. Welche Funktionen dies genau waren (warscheinlich nur einige Bug-fixes), konnten wir in unseren Tests nicht herausfinden, eventuell wird uns Commodore bezüglich dieses Punktes "aufklären".

Mit "Version" kann man nun nicht mehr nur die Versionsnummer von Kickstart und Workbench anzeigen, sondern auch die von Libraries (z.B."version graphics.library"), was recht nützlich sein kann.

Die Drucker-Treiber

Einer der Punkte, dem wir in der Redaktion am gespanntesten entgegensahen, waren die neuen Druckertreiber, die, laut Gerüchteküche, "sehr gut" und "furchtbar schnell" werden sollten. Doch ein "Dir devs:printers" rief erst einmal ein enttäuschtes "Oh" bei allen Umstehenden hervor, denn dort befand sich kein Treiber, der schon vom Namen her versprach, die NEC-Pinwriter-Drucker ordentlich anzusteuern. Es gibt dort lediglich einen Treiber für MPS1000, HP Laserjet, HP Paintjet, Okidata 2931, Xerox 4020 und zwei Epson-Treiber, einen für die "X"- und einen für die "Q"-Serie (auf Deutsch: 8/9 und 24 Nadeln). Zwecks eines Vergleichs verwendeten wir den "Butcher", um mit ihm ein HAM-Bild (320x200) und ein Hires/Interlace-Bild (Ford Model T) auszudrucken. Zuerst taten wir dies mit dem "normalen" Printer.device und dem Treiber CBM_MpS2xxx, mit dem wir auf folgende Zeiten kamen: Das HAM-Bild war nach etwa 30:40 Minuten fertig, das Hires-Bild (Grey-Scale) nach 23:30 Minuten. Danach kopierten wir das Printer.device sowie den "EpsonQ"-Treiber auf die Butcher-Disk und starteten den Versuch neu. Hierbei kamen wir dann zu neuen, erstaunlichen Zeiten: Das HAM-Bild war nach 6:50 Minuten fertig, was einem Faktor von 4,5 entspricht. Das Hires-Bild war nach sage und schreibe 94 Sekunden fertig, eine Steigerung um den Faktor 14,6.

Wenn auch das Einschaltbild nicht viel neus verrät, so verbirgt es doch einige Überraschungen



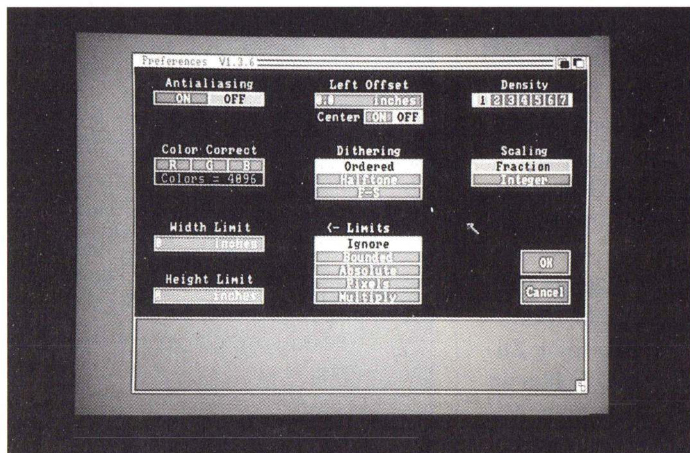
Ein Blick in das "devs"-Directory offenbart, wo diese Geschwindigkeitssteigerung vermutlich verankert ist: das Printer.device ist ca. 10 K länger geworden, was sicherlich nicht auf zufällige Mutation zurückzuführen ist. Alles in allem sind das nun Zeiten, die zu einem Rechner mit 68000-Prozessor passen...

Das L-Directory

Auch hier hat sich einiges getan. Gleich sind jedoch der Port-Handler und der Disk Validator geblieben, der Ram-Handler erfuhr einige Veränderungen, deren genaue Inhalte bisher noch nicht zu erfahren waren. Dagegen gibt es hier eine Reihe neuer Files, wobei an erster Stelle ein neuer "CON"-Handler mit dem Namen "NewCON" zu nennen wäre. Dieser verhilft jedem CLI bzw. der Shell zu einer einfachen History. Des weiteren kann man jetzt auch mit den Cursortasten korrigieren, ohne Zeichen löschen zu müssen.

Ein weiters neues Files im L-Directory ist das "FastFileSystem", das für schnelleren Zugriff auf HardDisks zuständig sein dürfte (laut Andy Finkel bis 600k/s, im Test brachte es eine "langsame" 30Meg-Disk (66 ms) mit Omti-Controller auf ca. 260k/s). Genaueres werden wir auch hier in einer späteren Ausgabe bekanntgeben. Es fiel im Test auf, daß noch einige Fehler enthalten sind, die hoffendlich bald entfernt sein werden.

Interessant ist auch, daß nun "serienmäßig" ein Pipe-Handler mitgeliefert



Auch im Graphic 2-Teil gibt es Änderungen, die mit einigem Probieren entschlüsselt wurden.

wird, mittels dem Tasks einfach kommunizieren können; so kann man z.B. mit "type demo.txt pipe:" von einem CLI einen Text zwischenspeichern und mit "type pipe:#?" dann von einem anderen CLI ausgeben. Dabei stecken natürlich in dem Pipe-Handler noch ganz andere Möglichkeiten, auf die wir hier verständlicherweise nicht eingehen können.
Weitaus interessanter ist da eine neue RAM-Disk mit dem Namen "Card:", die nicht nur resetfest ist, sondern von der man auch booten kann! Dabei kann man sogar, bei genügend Speicher, die RAM-Disk auf 880k einstellen, eine Diskette mit DiskCopy (!) hineinkopieren und dann mit hoher Geschwindigkeit booten.

Utilities

Hier haben sich eigentlich nur zwei neue Programme "eingeschlichen", die beide von diversen PD-Disks bekannt sind, und zwar sind das "ClockPtr", der die Maus in eine digitale Uhr verwandelt, und "CMD" , das die Ausgabe auf "Par:" oder "Ser:" in ein File umleitet.

Preferences

Etwa zwei Tage vor dem Redaktionsschluß erhielten wir weitergehende Informationen, so daß dieser Abschnitt praktisch total neu geschrieben wurde. Wie vorher berichtet, haben die neuen Druckertreiber sich auf Geschwindigkeitsseite stark verbessert, doch können sie vielleicht noch mehr? Ein Blick auf Preference-Druckereinstellung (Bild 1) genügt, dann sieht man die Änderungen. Ein neues Menü ist hinzugekommen, und auch einige programmtechnische Änderungen wurden vorgenommen. Im Graphic 1-Teil (der alte Graphic-Teil) wurde neben GrayShade noch ein zweiter Grauschattierungsmodus hinzugefügt, dessen Ergebnisse allerdings nicht überzeugen. Im Graphic 2-Teil (Bild 2), der komplett neu ist, erwarten den Benutzer einige freudige Überraschungen, deren Bedeutung wir teilweise im "Trial-and Error"-Verfahren herausfanden, teilweise den Ausführungen von Dave Berezowski, einem der Commodore-Programmierer, entnahmen.

- Antialiasing
Ist dieser Modus gesetzt, werden vor der Ausgabe auf dem Drucker noch Zwischenpunkte berechnet, was zwar den Ausdruck in etwa um den Faktor 2 verlangsamt, jedoch (im Test mit unserem 24-Nadler) auch bei normalen Ausdrucken zu einer wesentlich besseren Qualität führt. So hatte ein Ausdruck mit PageSetter endlich die Qualität, die man von einem DTP-Programm erwartet.

- Left Offset / Center
Hiermit kann man den Ausdruck auf dem Papier verschieben bzw. zentrieren.

- Density
Mittels Density bestimmt man die Druckdichte. Dabei entspricht "1" dem normalen 8-Nadel-Ausdruck, während mit steigender Dichte mehr und mehr Zwischenpunkte gesetzt werden. Dabei werden Druckdichten, die Programme selber setzen, überschrieben, so daß man seine Graphiken so gut ausdrucken kann, wie man selber will, und nicht, wie das Programm "will".

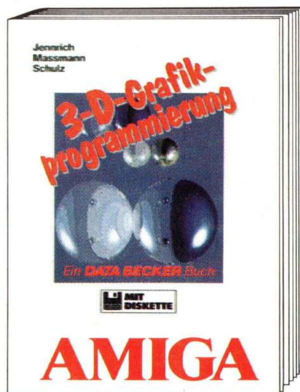
- Color Correct[ion]
Diese Option sorgt dafür, daß man bei einem Ausdruck einzelne Farbanteile verringern kann, so daß der Ausdruck "wirklichkeitsgetreuer" wird. Meistens muß man nämlich den Rot-Anteil verringern und Gelb verstärken, um die "Schwächen" des Farbbandes auszugleichen.

- Dithering
Dieses Menü bestimmt, wie die Übergänge schattiert werden. Dabei stellt "Ordered" den Default-Wert dar, bei dem alles "geordnet" schattiert wird. "Halftone" eignet sich am besten für Drucker mit einer Auflösung über 150 DPI, da hier ein Verfahren angewandt wird, das dem der Zeitung ähnlich ist und hierfür eine höhere Auflösung benötigt wird (z.B. NEC Pinwriter oder andere 24-Nadel-Drucker). Die "F/S"-Schattierung basiert laut Dave auf der "Floyd-Steinberg"-Methode der Fehler-Verteilung. Leider hat er nicht erklärt, was dabei genau passiert, es fällt nur auf, daß der Ausdruck etwa um den Faktor 2 langsamer wird und das Bild fast einem etwas schlechteren S/W-Foto ähnelt...Leider funktioniert diese Option nicht zusammen mit "AntiAliasing", da dann eine noch bessere Qualität zu erreichen wäre.

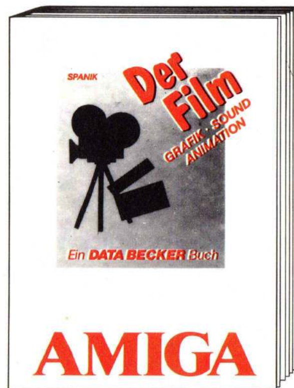
- Scaling
Fraction: Normale Skalierung (Default-Wert).
Integer: Wird dieses Flag gesetzt, sorgt der Treiber dafür, daß jeder Punkt auf dem Bildschirm als eine gerade Anzahl Punkte ausgedruckt wird, wobei das dann von der maximalen Auflösung des Druckers abhängt. Diese Option sollte man setzen, wenn man Text-Hardcopies oder Bilder mit dünnen vertikalen und horizontalen Linien ausdrucken will.



RUND UM DEN

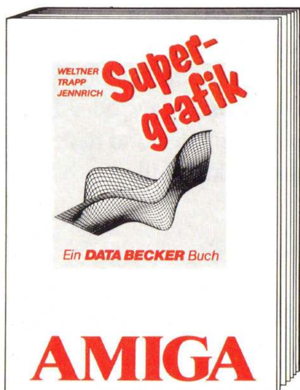


Licht und Schatten.
Dieses Buch öffnet Ihnen den Weg in eine faszinierende Welt. Anhand ausgefeilter Grafikalgorithmen erzeugen Sie phantastische Bilder: naturgetreue Spiegelungen, Licht und Schatten. Alles absolut realistisch und automatisch berechnet. In allen Auflösungen mit bis zu 4096 Farben!
Amiga 3-D-Grafikprogrammierung
Hardcover, 283 Seiten
inkl. Diskette, DM 59,–



Filme aus dem Amiga?
Dieses Buch macht's möglich. Vom Drehbuch bis zum fertigen Film. Mit allem, was dazu gehört: Grafik, Sound, Animation und und und. Dabei arbeiten Sie mit den verschiedenen DeLuxe-Programmen und lernen alles über den professionellen Einsatz von Sounddigitizer und Videorecorder.
Amiga – Der Film
Hardcover, 407 Seiten, DM 49,–

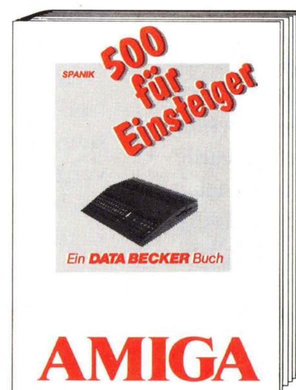




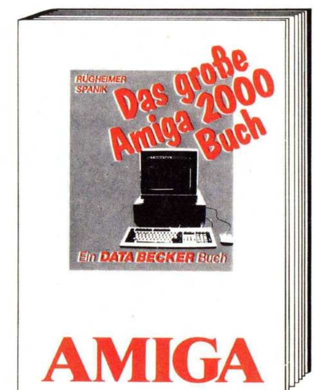
Das Buch zum Thema Nr. 1.
Grafik auf dem Amiga – mit dem entsprechenden Know-how ist hier fast alles möglich. Dieses Buch bietet es Ihnen: Nutzung der Libraries, die Register der Grafik-Chips, Aufbau und Programmierung von Screens, Windows, Halfbrite und Interlace aus BASIC und C uvm.
Amiga Supergrafik
Hardcover, 686 Seiten
inkl. Diskette, DM 59,–



Alles zur Amiga-Floppy.
Wie umfassend die Informationen eines Floppy-Buches von DATA BECKER sind, dürfte ja wohl bekannt sein. Im Amiga Floppybuch finden Sie darüber hinaus noch eine ganze Menge mehr, z. B. ein Superkopierprogramm, einen Floppyspeeder, einen Diskmonitor...
Amiga Floppybuch
Hardcover, 398 Seiten
inkl. Diskette, DM 59,–



Gleich loslegen.
Bei „500 für Einsteiger" heißt es aufstellen, anschließen und sofort loslegen. Schnell und für jedermann verständlich zeigt Ihnen dieses Buch, was Sie mit Ihrem neuen Rechner so alles anstellen können. Workbench, AmigaBASIC, CLI und DOS – schon bald wissen Sie, worauf es ankommt.
Amiga 500 für Einsteiger
343 Seiten, DM 39,–

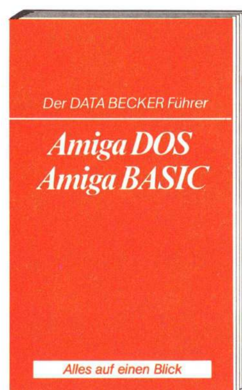


Ihr ständiger Begleiter.
Haben Sie einmal mit dem großen Amiga-2000-Buch gearbeitet, wird es sicherlich seinen festen Platz neben Ihrem Amiga behalten. Denn dieses Buch bietet Ihnen mehr als eine detaillierte Einführung. Vom Laufwerkseinbau bis hin zum Kickstart im RAM wird hier echtes Profi-Wissen vermittelt.
Das große
Amiga-2000-Buch
Hardcover, 684 Seiten, DM 59,–

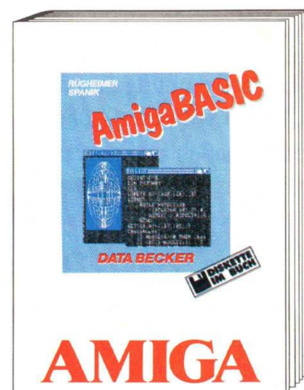
DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

AMIGA!

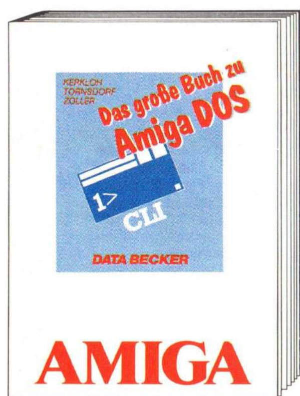




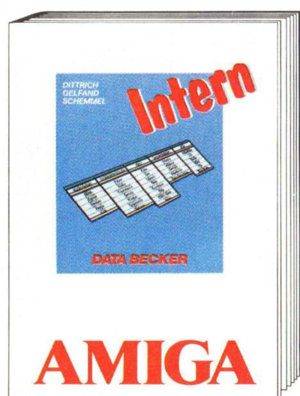
Alles auf einen Blick.
Der DATA BECKER Führer zu AmigaDOS und AmigaBASIC – das schnelle Nachschlagewerk für jeden Amiga-Anwender. Alle Befehle und Kommandos finden Sie hier auf einen Blick.
Der DATA BECKER Führer zu AmigaDOS & -BASIC
269 Seiten, DM 24,80



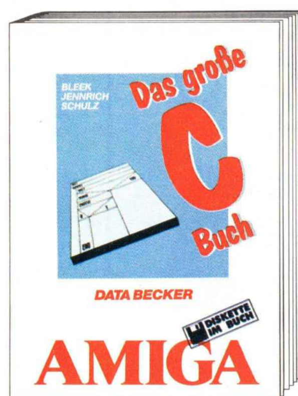
AmigaBASIC komplett.
Dieses Buch zeigt auf über 770 Seiten, worauf es beim Programmieren in AmigaBASIC ankommt. Natürlich mit jeder Menge interessanter Programmbeispiele, die auch gleich auf Diskette mitgeliefert werden. Ebenfalls im Buch: eine detaillierte Beschreibung des AC-BASIC-Compilers.
AmigaBASIC
Hardcover, 775 Seiten
inkl. Diskette, DM 59,–



Runter von der Workbench.
Rein ins AmigaDOS. Denn hier eröffnen sich ganz neue Möglichkeiten: Umlenken von Ein- und Ausgabe, mit RAM-Disk und CLI arbeiten, eigene CLI-Befehle programmieren, Batch-Dateien, Multitasking mit dem CLI, STARTUP-Sequenz... Das Know-how hierzu finden Sie im großen Buch zum AmigaDOS.
Das große Buch zu AmigaDOS
Hardcover, 310 Seiten, DM 49,–



Know-how aus erster Hand.
Amiga Intern – der ganze Rechner komplett in einem Buch: 68000-Prozessor, CIA, Blitter, Customchips, die Strukturen von EXEC, I/O-Handhabung, Verwaltung der Ressources, EXEC-Base, resetfeste Programme, IFF-Format, Programmierung der EXEC- und DOS-Routinen... Eben ein typisches Intern von DATA BECKER.
Amiga Intern
Hardcover, 639 Seiten, DM 69,–



Profi-Programme in C.
Wenn Sie an C Spaß gefunden haben, gibt Ihnen das große C-Buch den letzten Schliff. Denn hier erfahren Sie nicht nur, wie ein C-Compiler arbeitet und wie Sie selbst die schwierigsten Probleme in C lösen, sondern auch, wie Sie eine optimale Benutzeroberfläche entwickeln.
Das große C-Buch zum Amiga
Hardcover, 682 Seiten
inkl. Diskette, DM 69,–

Coupon!

HIERMIT BESTELLE ICH

NAME, VORNAME

STRASSE

ORT

zzgl. DM 5,– Versandkosten unabhängig von der bestellten Stückzahl
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

- Limits
Mit diesen drei Menüs kann man ganz genau bestimmen, wie groß der Ausdruck werden soll, wieviel Punkte in X- und Y- Richtungen gedruckt werden usw. Dabei haben die einzelnen Gadgets folgende Bedeutung/Auswirkung:
Ignore: Flag, um die Kompatibilität mit den alten Anwendungen zu gewährleisten. Dabei wird die Größe des Bildes folgendermaßen berechnet:

```
Breite = (rechterRand-
          linkerRand+1)
          Zeichen_pro_Zeile
Höhe =    Zeilen_pro_Seite/
          Zeilen_pro_Zoll
```

- Bounded
Mit dieser Option stellt man die maximale Grafik-Größe auf die Werte der beiden String-Gadgets; d.h., wenn man das Bild maximal 4 * 5 Zoll groß haben will (es aber kleiner sein kann!), dann stellt man die Werte auf 4.0 bzw. 5.0 und setzt das Bounded-Flag.

- Absolute
Ist dieses Flag gesetzt, wird das Bild genau so groß gedruckt, wie es die beiden Werte angeben, was dazu führen kann, daß das Bild verzerrt wird. Läßt man dabei einen der Parameter offen, wird das Bild entsprechend dem anderen Parameter ausgedruckt.

- Pixel
Entsprechend dem "Absolut"-Flag werden hier auf dem Drucker genausoviele Pixel ausgedruckt, wie in den Parametern angegeben wird. Dabei ist zu beachten, daß dies dann auch abhängig von der eingestellten Druckdichte ist, so daß ein Bild, das mit Density 1 richtig proportioniert ist, mit Density 7 viel zu schmal wird.

- Multiply
Diese Option führt dazu, daß x-und y-mal die Anzahl der Pixels im Bild gedruckt werden. Stehen die beiden Parameter z.B. auf 3 und 2 bei einem 320*200-Bild, so wird es mit 960*400 Pixels ausgedruckt.
Dies war zwar noch nicht alles, was in Kickstart und Workbench geändert wurde, doch läßt sich schon erahnen, daß die neuen Optionen die Amiga-Power noch verstärken. Es bleibt zu hoffen, daß die fertige neue Kick und Workbench V 1.4 bald erhältlich sein werden und daß dann noch mehr Erweiterungen enthalten sind. Zur Zeit bemühen wir uns um ein Interview mit Dale Luck, dem Grafik-Programmierer des Amiga bei Commodore, von dem wir hoffentlich noch mehr über das neue Double erfahren werden.

ENDE

JÜRGEN SCHMIDT

PHOTON PAINT

Konkurrenz für DPAINT ?

Die besonderen grafischen Fähigkeiten des Amiga sind nach wie vor unbestritten. Kein Wunder also, daß immer mehr Mal- und Zeichenprogramme zu den schon existierenden Produkten auf den Markt drängen. Photon Paint, eines der neuesten Produkte auf diesem Sektor, bietet genug interessante Features, um einmal etwas näher behandelt zu werden.

Das Wichtigste an Mal- und Zeichenprogrammen ist, von den Tools einmal abgesehen, wohl die Anzahl der verfügbaren Farben. Da Photon Paint alle Darstellungen ausschließlich im HAM-Modus verarbeitet, können bis zu 4096 Farben gleichzeitig für eine Darstellung verwandt werden. Dabei kann das Programm entweder in den PAL-Auflösungen 320 x 256 oder 320 x 512 (interlaced) sowie im Overscan-Modus 352 x 282 bzw. 352 x 564 (interlaced) oder dem NTSC-Standard gefahren werden. Das eigentliche Programm meldet sich mit einer Toolbox, die neben der aktuellen Palette alle üblichen Standardfunktionen wie verschiedene Brushes, einfache und gefüllte Rechtecke, Kreise, Ellipsen und Polygone sowie Linien beinhaltet.

Weitere Funktionen sind die vergrößerte Darstellung eines Bildes in einem separaten Fenster, dessen aktueller Bereich frei verschiebbar ist, das Zurücknehmen von Veränderungen mit Undo sowie die Möglichkeit der Texteingabe mit den verschiedensten Fonts. Drei verschiedene Füll-Modi stehen dem Anwender zur Verfügung: Mit Background, Boundary und Free Shape kann zwischen Hintergrundfarbe, Umrandung und frei zu zeichnendem Bereich unterschieden werden, inwieweit eine Fläche mit einer bestimmten Farbe gefüllt werden soll.

Außergewöhnliche Farbvielfalt

Um zwischen 4096 Farben auswählen zu können, muß das erweiterte Paletten-Fenster eingeschaltet werden. Die möglichen Farbschattierungen sind über die RGB- und/oder HSV-Schieberegler sowie durch direktes Anklikken der Farbe - in den Farbboxen oder dem Bild selbst - einstellbar. Mit Swap werden zwei Farben untereinander ausgetauscht, Copy kopiert eine Farbe in eine andere. Spread bestimmt einen Farbverlauf zwischen zwei Farben der Palette. Eine besondereres Feature bietet die Palette, denn sie stellt 64 Farben gleichzeitig zur Verfügung. Eine Veränderung einer Farbe der Palette kann, je nach Einstellung, mit oder ohne Auswirkung auf das Bild geschehen. Mit der Exact-Color-Funktion sind exakte Farbabgrenzungen ohne fließende Übergänge möglich. Diese Funktion ist wichtig, da im HAM-Modus die Zeichenfarbe normalerweise von dem in der Darstellung angrenzenden Pixel beeinflußt wird. Die gerade aktuellen Palettenfarben können von verschiedenen Quellen wie z.B. der Brush, einem soeben eingeladenen Bild oder dem Reservebildschirm stammen. Die Toolbox ist auf dem Screen frei verschiebbar und kann je nach Bedarf ein- oder ausgeschaltet werden.

Die Menüs

Die Titelleiste von Photon Paint enthält fünf verschiedene Pull-Down-Menüs. Mit dem Project-Menü lassen sich Bilder und Brushes laden, speichern, drucken sowie löschen. Das Programm verfügt dazu über einen angenehm zu bedienenden und schnell arbeitenden File-Selector, der einen eigenen Screen benutzt und daher immer in derselben Auflösung erscheint. Wird ein File geladen, der eine andere Auflösung als der gerade aktuelle besitzt, so kann das Programm, ohne den Vorgang abbrechen zu müssen, direkt dazu veranlaßt werden, den Bildschirm automatisch anzupassen und den File daraufhin zu laden.

Mit der Swap-Option kann auf einen zweiten im Hintergrund liegenden

Screen geschaltet werden, um z.B. Bilder eines anderen Malprogramms oder Teile daraus auszuschneiden und in eigenen Darstellungen zu verwenden.
Das Font-Menü ermöglicht das Laden und Auswählen verschiedener Fonts und unterschiedlicher Schriftattribute wie Italic, Bold, Underline, Outline und Reverse.
Mit Change Disk können selbst erstellte oder aus anderen Programmen importierte Fonts eingelesen und benutzt werden.

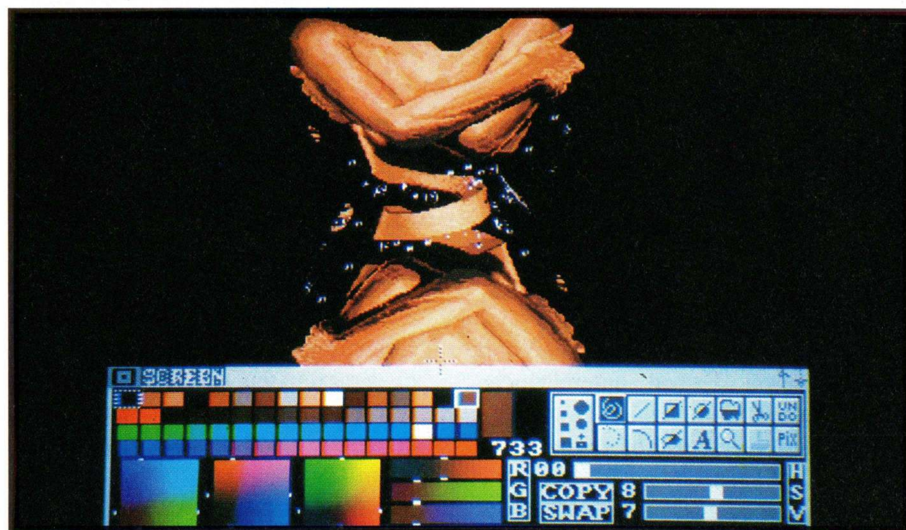
Umfangreiche Tools

Seine große Stärke spielt das Programm jedoch erst bei den Brushes aus. Neben den Standardfunktionen wie horizontales und vertikales Spiegeln (Flip), beliebiges Vergrößern (Resize) und Rotieren (Rotate), Verbiegen (Bent) oder dem perspektivischen Kippen entlang der Achsen einer Brush (Tilt) können diese zusätzlich um verschiedene dreidimensionale Körper wie Zylinder, Kegel, Ellipse, Kugel oder einen Würfel gewickelt werden. Eine eigene Form kann ebenso erstellt werden wie eine dreidimensionale Spiralform. Größe und Form der verschiedenen Körper sind mittels der Maus frei auf dem Bildschirm einstellbar und können durch die Anzeige der Koordinaten in der Toolbox ständig kontrolliert werden. Der Vorteil anderen Programmen wie z.B. Digipaint gegenüber liegt klar auf der Hand: Photon Paint stellt umfangreiche Tools zum perspektivischen Zeichnen bereit, die auch Ungeübten das einfache Erstellen räumlicher Zeichnungen ermöglicht.

Spot an!

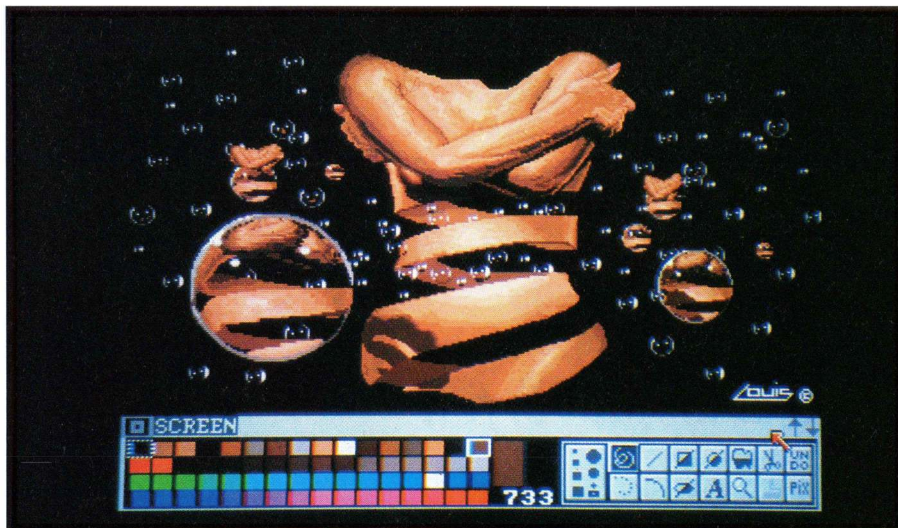
Zusätzlich kann eine Lichtquelle in die Darstellung jeder beliebigen Brush eingebracht werden. Die Lichtquelle kann dabei um das Objekt herum frei im Raum plaziert werden und hat je nach Einstellung der Intensität unterschiedliche Auswirkungen auf die Darstellung. Auf diese Art können sehr sanfte oder auch harte Farbschattierungen erreicht werden. Die Transparent-Funktion sorgt dafür, daß die Hintergrundfarbe der Brush nicht auf den Screen übertragen wird.





Mischen mit dem Hintergrund

Das Mode-Menü andererseits wacht darüber, wie der Inhalt einer Brush auf den Hintergrund gesetzt wird. Normalerweise ersetzt die Brush den Hintergrund. Mit ADD und SUBTRACT stehen zwei logische Operationen zur Verknüpfung von Hintergrund und einer Brush zur Verfügung. Mit Blend wird eine Brush mit dem Hintergrund kombiniert. Mit SET kann nun eingestellt werden, inwieweit sich die Hintergrundfarbe auf die Darstellung der Brush auswirkt bzw. die Brush den Hintergrund überdeckt. Diese Einstellungen werden in zwei Edierfenster eingezeichnet und können in einem weiteren Fenster sofort kontrolliert werden. Mit DITHER wird ein mehr oder weniger verschwommener Übergang zwischen den Farbabstufungen der Set-Einstellungen erreicht. Je nach Einstellung von DITHER lassen sich dabei wirklich sehr fließende Übergänge erzeugen, die auch beim Zeichnen von Rechtecken und Kreisen, egal ob gefüllt oder nicht, sowie bei einfachen Linien auftreten. Die SET-Einstellungen lassen sich durch die Wahl des NORMAL-Modus wieder abschalten, so daß dem Anwender sofort eine gebrauchsfähige Brush zum einfachen Kopieren von Bildern oder Bildausschnitten zur Verfügung steht, ohne daß die SET-Einstellungen dabei verloren gehen. Dazu sollte noch gesagt werden, daß Photon Paint keine extra CLIP-, CUT- oder COPY- Funktionen beinhaltet; diese Aufgaben können auch mit der Brush erledigt werden, was zusätzlich die Möglichkeit bietet, eventuelle Veränderungen sofort durchzuführen. Ein weiterer Vorteil dieser Methode ist der Umstand, daß die Brush nicht an die herkömmliche rechteckige Form gebunden ist, so daß



jede beliebige unregelmäßige Form aus einem Bild ausgeschitten werden kann.
Durch Kombination der verschiedenen Techniken wie Wickeln einer Brush um einen dreidimensionalen Körper sowie der Benutzung der BLEND-Einstellungen und einer Lichtquelle sind realistisch wirkende, dreidimensionale Darstellungen recht einfach zu realisieren. Einige Beispiele sind das Erstellen spezieller Fonts von beliebiger Größe, die mit 3D- und Lichteffekten modifiziert werden können, oder das Erzeugen von recht realistischen wirkenden Kugeln, die mit verschiedenen Füllmustern belegt wurden.
Die vielen Möglichkeiten, die sich dem Anwender durch geschicktes Verarbeiten der Brushes bieten, können hier, ohne den Rahmen dieses Tests zu sprengen, nicht alle im einzelnen beschrieben werden. Hier spiegeln sich aber die Vorteile von Photon Paint anderen Programmen gegenüber wieder. Positiv fiel in diesem Zusammenhang auch die hohe Verarbeitungsgeschwindigkeit bei praktisch allen Operationen auf.
Sollen alle Features von Photon Paint voll ausgeschöpft werden, so sollten minimal 1 Megabyte Speicher sowie ein zweites Diskettenlaufwerk vorhanden sein. Stehen nur 512 Kbyte Speicher zur Verfügung, so ist Photon Paint nur eingeschränkt nutzbar. Dies betrifft vor allem das Arbeiten mit Interlace-Darstellungen oder zwei Screens sowie dem Undo-Speicher.
Das Software-Paket besteht aus einer Programm- und einer Daten-Diskette mit einigen sehr gefälligen Bildern. Das leider nur in englischer Sprache vorliegende Handbuch erklärt alle Features von Photon Paint ausreichend und gibt einige praktische Beispiele und Tips, mit deren Hilfe die verschiedenen Funktionen des Programms schnell erlernbar sind, immer vorausgesetzt, man ist der englischen Sprache mächtig. Dieser Umstand gewinnt besonders an Bedeutung, da einige Features nicht direkt auf dem Bildschirm erkennbar sind. Die Bedienung des Programms erschien daher am Anfang auch etwas ungewohnt, stellte sich im Laufe der Zeit aber doch als recht angenehm heraus.

Fazit

Obwohl Photon Paint die meisten Features anderer Malprogramme beinhaltet, so mußte doch auf einige Optionen wie Colorcycling oder das Erstellen von eigenen Füllmustern verzichtet werden. Ein weiteres Manko ist die Tatsache, daß mit den Brushes von Photon Paint nicht gemalt werden kann. Da diese Features aber von Programmen wie DeluxePaint oder DigiPaint, die wohl jeder Amiga-Anwender kennen dürfte, unterstützt werden und Photon Paint das Einladen von Files aus anderen Programmen unterstützt - hier ergaben sich fast keine Beschränkungen - kann Photon Paint auch als sinnvolle Ergänzung zu Programmen wie DeluxePaint oder DigiPaint angesehen werden. Eine Einschränkung bei der Zusammenarbeit der verschiedenen Programme muß allerdings gemacht werden: Files, die mit Photon Paint abgespeichert wurden, sind von Deluxe Paint wegen des HAM-Modus nicht lesbar.
Durch die mannigfaltigen Möglichkeiten des perspektivischen Zeichnens mit den Brushes von Photon Paint und des Verabeitens verschiedener Bildformate im HAM-Modus sowie die exakte Verarbeitung von 4096 Farben gleichzeitig muß das Programm jedoch mit in die Spitzenklasse unter den Mal-Programmen eingeteilt werden, zumal alle Standardfunktionen zum Zeichnen implementiert wurden.

Photon Paint

+ Verarbeitung im HAM-Modus
+ 64 Farben in der Palette
+ Perspektivische Projektion der Brushes
+ Reformatierung des Bildschirms ohne Bildverlust
+ Hohe Verarbeitungsgeschwindigkeit
+ Simulation einer Lichtquelle

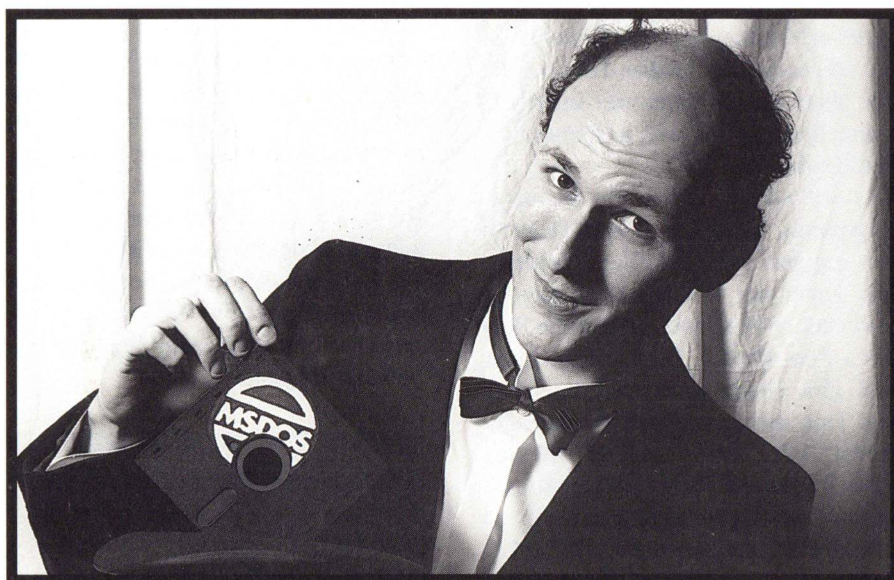
- Kein Füllen mit Patterns
- Kein Zeichnen mit Brushes
- Englisches Handbuch

Hersteller: MicroIllusions
Anbieter: IM, Fachhändler

VON CHRISTIAN KELLER

PC-INTIM

Teil 4: Diskuswerfen



Heutiges Thema von PC-Intim wird alles sein, was mit Disketten zu tun hat. Ein Einblick in die verschiedenen Diskettenformate soll gewährt werden, ebenso werden wir auf die Diskettentreiber eingehen.

Was wäre ein Computer ohne Massenspeicher? Ohne elektrische Energieversorgung würde er sehr schnell alles vergessen, was man ihm mit viel Mühe eingegeben hat. Ein Speichermedium, das auch nach Ausschalten der Spannungsversorgung die Daten erhalten kann, ist also unverzichtbarer Bestandteil einer sinnvoll nutzbaren Rechenanlage. In den Ursprüngen der Datenverarbeitung benutzte man dazu Magnetbänder. Die meisten unter Ihnen werden diese Art der Datensicherung aus den ersten Zeiten mit dem C64 kennen. Nachteil des Ganzen ist aber die lange Zugriffszeit. Um dem Manko dieser sequentiellen Sicherungsart zu begegnen, kam ein schlauer Kopf auf die Idee, die Daten in konzentrischen Ringen auf eine Magnetscheibe zu schreiben. Damit war die Diskette erfunden. Heutzutage haben die Disketten in allen Bereichen die Magnetbänder verdrängt und sind wohl das Speichermedium schlechthin.

Neuerdings erhält die Diskette aber Konkurrenz von der Festplatte, die mit einer wesentlich höheren Kapazität und einer schnelleren Zugriffszeit glänzen kann. Vom Prinzip her sind aber beide Speicher gleich: Magnetscheiben mit beweglichen Schreib-Leseköpfen, welche die Daten in Kreisen auf das Speichermedium schreiben. Im Gegensatz zur Floppy gibt es aber bei einer Festplatte keine austauschbaren Datenträger, so daß Probleme mit der Kompatibilität beim Austausch von Daten gar nicht erst auftreten können. Außerdem gibt es bei Festplatten nur ein Format, was bei Disketten nicht der Fall ist.

Ein Herr von Format

Unter MS-DOS gibt es, im Gegensatz zum AMIGA-DOS, eine ganze Reihe von verschiedenen Diskettenformaten. Diese entstanden hauptsächlich aufgrund der steigenden Qualität von Disketten und der dazugehörigen Hardware (sprich Floppy-Drives). Standardgröße der Floppys bei IBM war das 5,25-Zoll-Format, das mit 40 Spuren beschrieben wurde. Verwendete man anfangs noch ein 8-Sektoren-Format, so ging man bald zu den auch heute noch gebräuchlichen 9-Sektoren-Formaten über. Daraus resultieren die vier heute von allen Standard-PC's schreib- und lesbaren Diskettenformate:

Einseitig, 8 Sektoren, 160 kByte
Zweiseitig, 8 Sektoren, 320 kByte
Einseitig, 9 Sektoren, 180 kByte
Zweiseitig, 9 Sektoren, 360 kByte

Diese Formate sind von allen DOS-Versionen lesbar. Des weiteren gab es noch einige MS-DOS-Rechner, die ihre Disketten mit 80 Spuren beschrieben. Diese hatten dann jeweils den doppelten Speicherplatz auf der Diskette zur Verfügung. Diese 80-Spur-Formate nehmen aber eine Sonderstellung ein, die bei der Einführung der 3,5-Zöller (die für den Rechner identisch mit 80-Spur-5,25-Zoll-Laufwerken sind) zu einigen Problemen führte. Nächster Schritt in der Weiterentwicklung war die Einführung der HD(High Density)-Disketten. IBM führte diese gemeinsam mit den Rechnern der AT-Serie ein. Durch ein anderes Magnetisierungsverfahren war es nun möglich, auf demselben Platz mehr Daten abzulegen. Bei diesen Disketten war eine Spur mit 15 Sektoren beschrieben, außerdem wurden 80-Spur-Laufwerke verwendet. So kam man auf eine Kapazität von 1200 kByte (bei diesem Format wird grundsätzlich zweiseitig beschrieben). Diese Disketten sind nur mit den speziellen Laufwerken der AT's schreib- und lesbar. Ebenso ist eine DOS-Version ≥ 3.0 notwendig.
Mit der weiteren Verbreitung der 3,5-Zoll-Laufwerke hielten auch diese Einzug in die IBM-Welt. In den neuen Modellen von IBM werden sie sogar schon serienmäßig eingebaut. Im Gegensatz zu Atari und AMIGA werden hier aber zwei verschiedene Laufwerkssarten unterstützt. Zuerst wären da die normalen Laufwerke zu nennen. Sie werden mit 80 Spuren, 9 Sektoren und zwei Seiten formatiert und bieten so 720 kByte Speicherkapazität. Diese Disketten sind vom physikalischen Aufzeichnungsverfahren identisch mit den AMIGA-Floppys. Zusätzlich unterstützen die PS/2-Modelle ein 3,5-Zoll-HD-Format. Folgende Parameter sind für sie gültig: 80 Spuren, 18 Sektoren, zweiseitig. So kommt die recht große Kapazität von 1,44 MByte zustande. Dieses Format ist aber nur mit einem speziellen Floppy-Controller und den entsprechenden Laufwerken verwendbar. Da die 3,5-Zoll-Floppys bei einem normalen PC oder AT nicht vorgesehen sind, müssen sie extra eingebunden werden. Der Schlüssel zu diesen Laufwerken wie auch zu 5,25-80-Spur-Laufwerken ist jeweils ein spezieller Treiber.

Triebfedern

Wenn man mit MS-DOS > 3.0 arbeitet und ein 3,5-Zoll-Laufwerk sein eigen nennt, wird man bestimmt schon auf eine Eigenart gestoßen sein. Hat man eine auf 720 kByte formatierte Diskette, erkennt der PC diese ohne weiteres an und arbeitet mit ihr, ohne zu murren. Doch wehe, man versucht, diese Diskette zu formatieren: Beharrlich weigert sich der PC, die Diskette über die Spur 39 hinaus zu formatieren. Der Rechner kann zwar anhand des Media-Byte erkennen, um welches Diskettenformat es sich handelt, doch hat man vergessen, im System eine Möglichkeit zu implementieren, die es erlaubt, auch solche Disketten zu erzeugen. Dazu muß man zuerst einen Device-Treiber in das System einbinden. Dabei gibt es aber einen markanten Unterschied zwischen den MS-DOS-Versionen und PC-DOS. MS-DOS benötigt nämlich gar keinen externen Treiber. Bei MS-DOS ist eine zusätzliche Möglichkeit implementiert, um dem Rechner mitzuteilen, daß er 80-Spur-Laufwerke besitzt. Diese Möglichkeit bietet das Kommando DRIVPARM in der CONFIG.SYS-Datei. Mit diesem Befehl kann man die von DOS gesetzten Parameter der Block-Geräte (also Floppies, Festplatten usw.) überschreiben. Die Syntax des Befehls lautet:

```
DRIVPARM=/D:ddd /C /F:f /H:hh
        /S:ss /T:ttt
```

Dabei bedeuten die Parameter folgendes:

```
/D:ddd    'ddd' gibt die logische Laufwerksnummer an. Dabei ist A=0,
          B=1, usw.
/F:f      'f' ist ein Formfaktor für das Speichermedium.
          Einige Beispiele für f sind:

0         40-Spur-Laufwerk
1         AT-Laufwerk
2         80-Spur-Laufwerk
5         Festplatte
7         sonstiges Laufwerk

/H:hh     'hh' gibt die Anzahl der Schreib-Lese-Köpfe an.
/S:ss     'ss' spezifiziert die Anzahl der Sektoren pro Spur.


/T:ttt    'ttt' gibt die Anzahl der Spuren an.
/C        Dieser Schalter gibt an, daß die Laufwerksverriegelung
          unterstützt wird.
```

Wenn man ein Standardlaufwerk einbinden will, genügt es, den Formfaktor anzugeben. Hat man zum Beispiel ein externes AMIGA-Laufwerk an den PC-Teil angeschlossen, so muß man in CONFIG.SYS die Zeile

```
DRIVPARM /D:1 /F:2
```

einfügen. Mit den restlichen Parametern kann man die Konfiguration des Gerätes frei einstellen. Zu beachten ist dabei aber, daß die Gesamtzahl der Sektoren gleich bleibt. Daher ist für Floppys hier keine sinnvolle Konfiguration möglich.
Eine weitere Möglichkeit, die sowohl MS-DOS als auch PC-DOS bietet, ist die Konfiguration über den externen Gerätetreiber DRIVER.SYS. Dieser wird auch in die CONFIG.SYS-Datei eingetragen. Die Parameter sind identisch mit denen von DRIVPARM. Das Einbinden eines 720-kByte-Laufwerk lautet dann wie folgt:

```
DEVICE=DRIVER.SYS /D:1 /F:2
```

Hier können nun beliebige Konfigurationen eingegeben werden. Man hat also die Möglichkeit, sich eine Diskette mit z.B. 45 Spuren, einer Seite und 4 Sektoren anzulegen.
Was man aber beachten muß, ist die Tatsache, daß DRIVER.SYS ein neues logisches Laufwerk einrichtet. Besitzt man eine Festplatte, wird das 720-kByte-Laufwerk als D: eingerichtet, ohne Festplatte erhält es die Kennung C:. Daher ist es anzuraten, wenn mög-

lich das Kommando DRIVPARM zu benutzen, da man sich sonst sehr bald in den Regionen um X:,Y:und Z: bei den Laufwerksbuchstaben befindet, denn man hat ja auch noch die vier Janus-Drives.

Commodore-Spezialitäten

Besitzer einer SIDECAR werden noch ein MS-DOS 2.11 erhalten haben. Diese DOS-Version ist von Natur aus nicht in der Lage, 720-Kbyte-Disketten zu handhaben. Da Commodore einen Anschluß für die A1010 an der SIDECAR angebracht hat, war man auch so freundlich, die Möglichkeit des 80-Spur-Formatierens zu implementieren. Das erfolgt über den Aufruf

```
FORMAT B: /T
```

So formatierte Disketten sind auch mit DOS 3.x einwandfrei lesbar. Bei der Benutzung von DOS 3.x-formatierten 80-Spur-Disketten ergibt sich aber ein erhebliches Problem. Grund hierfür ist die etwas veränderte Diskettenstruktur von DOS 3.x-80-Spur-Floppys. Bei DOS > 3.0 wurden die beiden FAT's um einen Sektor verlängert, ebenso wurde die Clusterlänge auf 4 Sektoren erhöht. Dieses Format kann DOS 2.11 aber nicht lesen. Das 720er-Format von DOS 2.11 ist nämlich nur eine verlängerte 360er-Diskette, also von der logischen Struktur gleich. DOS 3.x hat mit diesen Disketten keine Probleme. Ein Datenaustausch auf 3,5-Zoll-Disketten zwischen Bridgeboard (3.2) und SIDECAR (2.11) ist also nur möglich, wenn die Disketten unter 2.11 formatiert wurden.

Um einmal etwas Klarheit in die Fachbegriffe im Bereich Disketten zu bringen, folgt hier nun eine kleine Erläuterung der Ausdrücke:

Spur

Die Oberfläche einer Diskette ist in konzentrische Ringe unterteilt, die sogenannten Spuren. Je nach Diskettenformat ist die Spurdichte unterschiedlich. Sie reicht von 48 TPI (Tracks Per Inch) bei 40-Spur-5,25-Zoll über 96 TPI bei 80-Spur-Disketten in 5,25 Zoll bis 135 TPI auf den 3,5 Zoll-Disketten.

Sektoren

Jede Spur ist noch einmal in Teile gesplittet. Diese Blöcke sind die kleinste separat speicherbare Einheit bei einer Diskette (daher auch der Begriff des "blockorientierten Gerätes"). Die Größe eines Sektors beträgt bei MS-DOS 512 Bytes.

Kapazität

Die Menge der möglichen gespeicherten Daten. Dabei muß man unterscheiden zwischen der unformatierten und der formatierten Speicherfähigkeit. Die unformatierte Speicherfähigkeit gibt an, wieviel Daten physikalisch auf die Diskette passen. Da aber jedes Betriebssystem Daten zum Erkennen von Sektoren und als "Lückenfüller" zwischen zwei Sektoren braucht, bleibt von der unformatierten Speicherfähigkeit nur ein Teil übrig, der jedoch von Rechnertyp zu Rechnertyp unterschiedlich sein kann. So beträgt die unformatierte Kapazität einer normalen 3,5-Zoll-Diskette 1 MByte. Formatiert man diese nun mit dem AMIGA, hat man die bekannten 880 kByte frei, unter MS-DOS hingegen bleiben nur 720 kByte übrig.

Cluster

Ein Cluster stellt einen Zusammenschluß mehrerer Sektoren dar, die nur gemeinsam alloziert werden können. Eine Datei belegt also immer einen Cluster (auch wenn sie nur ein Byte lang ist). Sinn der Cluster ist es, eine zu starke Zerstückelung von Dateien auf der Diskette zu verhindern. Bei den verschiedenen Formaten unter MS-DOS ist die Größe eines Clusters unterschiedlich. Die Clustergröße beträgt bei einseitigen Disketten 1 Sektor, bei zweiseitigen 40-Spur-Floppys 2 Sektoren und bei 80-Spur-Formaten 4 Sektoren, außer beim speziellen 80-Spur-Format von Commodores MS-DOS 2.11, wo auch 80-Spur-Disketten 2 Sektoren pro Cluster haben.

FAT

Die File Allocation Table ist ein Verzeichnis auf jedem Massenspeicher, in dem die Belegung der einzelnen Cluster angezeigt wird und außerdem die Verknüpfung der einzelnen Cluster zu einer ganzen Datei gespeichert ist.

Im nächsten Teil von PC-Intim wird auf die Disketten- und Festplattenverwaltung unter MS-DOS eingegangen. Dies ist auch für den AMIGA-Programmierer interessant, da der Interrupt 13h ja auch vom AMIGA benutzt wird, um auf eine Janus-Festplatte zuzugreifen.

ENDE

VON R. HOFMANN

GUT GEBACKEN

Die Software des Juniorprommer



Zum zweiten und letzten mal geht es hier um den Juniorprommer. Diesmal wird jedoch fast ausschließlich die Software besprochen. Über ein vielseitiges Assemblerprogramm, welches Sie in beliebige, eigene Programme einbinden können, sind alle gängigen Operationen mit Speicherbausteinen, wie z.B. Lesen, Programmieren, Vergleichen etc., möglich. Außerdem kann dieses Assemblerprogramm durch eine umfangreiche Konfigurationstabelle an die ausgefallendsten Speicherbausteine (auch Banking) angepaßt werden.

In der vorigen Ausgabe...

...wurde die Hardware und deren Programmierung beschrieben. Jetzt soll hauptsächlich ein Assemblerprogramm vorgestellt werden, welches aus Hochsprachen heraus benutzt werden kann, um alle Funktionen des Juniorprommers zu bedienen. Diese Assemblermodul hat den Namen Juniorsoft und wird hier in Verbindung mit einem C-Programm, genannt beispiel.c, vorgestellt. Die vorgestellten Beispiele sind generell auch auf andere Hochsprachen übertragbar.

Betrieb der Hardware an Amiga-Rechnern

Da die Hardware flexibel ausgelegt wurde, soll der Anschluß speziell an Amiga-Rechner noch einmal erläutert werden. Zuerst benötigen Sie ein Verbindungskabel zur 5V-Spannungsversorgung des Juniorprommers. Dabei können Sie entweder ein externes, stabilisiertes Netzteil verwenden, oder Sie greifen mit lediglich einer Leitung die Spannungsversorgung ihres Rechners am Joystick-Port ab. Dazu benötigen sie ein Verbindungskabel vom Joystickport-Pin 7 zum Mittelstift der Stromversorgungsbuchse des Juniorprommers. Die Steckbrücke zur Versorgung über die Schnittstelle dient nur zu Testzwecken und ist im normalen Betrieb nicht eingesteckt. Jetzt ist der Juniorprommer, sofern fehlerfrei auf-

gebaut, betriebsbereit. Um die Funktionsfähigkeit zu prüfen, ist das Programm "pruf.c" abgedruckt, mit dem die Funktionen der Hardware überprüft werden sollten. Leider hat sich seit der letzten Ausgabe noch ein Fehler in der Hardware gezeigt. Beachten Sie bitte die hier getrennt abgedruckte Korrektur.

Juniorsoft

In Listing 1 ist der Quellcode "junior.asm" in seiner ursprünglichen Version für den Aztec-Assembler abgedruckt. Adaptieren auf ein anderes Entwicklungssystem ist, falls nötig, nicht mit Schwierigkeiten verbunden, da weder Bibliotheken noch spezielle Funktionen des Entwicklungssystems benutzt werden.

1. Initialisieren des Juniorprommers für den weiteren Betrieb.
2. Bedienen der Schieberegister SRA bis SRD.
3. Bedienen des Adresszählers.
4. Kontrollieren der Datenrichtung.

Die weiteren Funktionen basieren auf den bisher genannten Funktionen.

5. Lesen des Speicherbausteins in den Speicher des Rechners.
6. Testen des Speicherbausteins, ob dieser leer ist (nur für EPROMs sinnvoll).
7. Vergleichen des Speicherbausteins mit dem Speicher des Rechners.
8. Prüfen, ob die Daten im Speicher des Rechners problemlos in den Speicherbaustein programmiert werden können (nur für EPROMs sinnvoll).
9. Programmieren eines Speichers.
10. Prüfen, ob ein Speicherbaustein des gewünschten Typs eingesetzt ist.
11. Speichertyp einstellen.
12. Initialisieren des Juniorprommers.

Das hier ausschließlich von "Speicherbausteinen" und nicht von EPROMs die Rede ist, hat folgenden Grund: Über eine flexible Konfigurationstabelle kann nahezu jeder Speicherbaustein mit 28 oder 24 Anschlüssen über den Juniorprommer bedient werden. So ist es z.B. auch möglich, batteriegepufferte SRAMs zu programmieren, exotische ROMs zu lesen oder EEPROMs zu programmieren. Listing 2 zeigt die Typentabelle, welche bereits die EPROMs von 2716 bis 27011 enthält. Wie sie neue Speichertypen definieren, wird noch erklärt. Es liegt in

```
; Junior Prommer Resourcen
; (c) Merlin Computer   KICKSTART
; erstellt am 20.3.88
; R.Hofmann
; Aufruf: Return = Junior (Funktion,P1,P2....);
;
; Aztec C 3.6
; as junior -N -C -D

max_f   equ 11

strobe  equ %01000000

azb     equ %00010000
lazb    equ %00000100

ciaapb  equ $bfe101
ciabpa  equ $bfd000
ciaapbd equ $bfe301

bpe     equ 82

; Indizies in die Typentabelle
; 82 Bytes/Eintrag

zkette  equ 0
kapazi  equ 30
seite   equ 34
alg     equ 38
i_zeit  equ 40
m_zeit  equ 42
trials  equ 44
standby equ 46
read    equ 48
odis    equ 50
program equ 52
pr_ver  equ 54
pr_inh  equ 56
page_wr equ 58
manufac equ 60
devcode equ 61
tsmask  equ 62
frei3   equ 63
frei1   equ 64
frei2   equ 66

m_a10   equ 68
m_a11   equ 70
m_a12   equ 72
m_a13   equ 74
m_a14   equ 76
m_a15   equ 78
m_a16   equ 80

        cseg
        public _Junior
        public _WaitForTimer

_Junior:
        movem.l d1-d7/a0-a6,save        ; Hauptverteiler
        lea 6(a7),a5                    ; Parameterbasis
        move.w -2(a5),d0
        cmp.w #max_f,d0
        bhi  f_err
        lsl.w #2,d0
        lea f_adr,a0

        movea.l 0(a0,d0.w),a0
        jsr  (a0)

ende:   movem.l save,d1-d7/a0-a6
        rts
f_err:  move.l $12345678,d0
        bra    ende
```

```
          dseg

; Zeiger auf Funktionen
f_adr:    dc.l srx                          ; 0=SRX
          dc.l srd                          ; 1=SRD
          dc.l az                           ; 2=AZ
          dc.l setzea                       ; 3=SETZEA
          dc.l lese                         ; 4=LESEN
          dc.l ltest                        ; 5=LTEST
          dc.l vergl                        ; 6=VERGL
          dc.l vtest                        ; 7=VTEST
          dc.l ipro                         ; 8=PROG
          dc.l tstest                       ; 9=TRTEST
          dc.l waltp                        ; 10=TYP
          dc.l init                         ; 11=INIT

          cseg

sra:      movem.l d0/d2,-(sp)
          moveq #2,d1
          or.b adrsra,d0
          move.w d0,-(sp)
          bsr srxd                          ; Byte herausschieben
          move.w (sp),d0                    ; alter Wert
          move.b d0,d1
          and.b #%11100000,d0               ; Testen auf Aenderung
          move.b letzt_a2,d2
          and.b #%11100000,d2
          cmp.b d0,d2
          beq  ta15                         ; Alles gleich
          scs d2
          move.w #1000,d0                   ; 0.1s Verzoegerung
          tst.b d2
          bne  zver
          move.w #8000,d0                   ; 0.8s Verzoegerung
zver:     bsr zeit
ta15:     and.b #1,d1                       ; A15 auf Aenderung
                                              testen
          move.b letzt_a2,d0
          and.b #1,d0
          cmp.b d0,d1
          beq  uspr                         ; Alles gleich
          moveq #50,d0                      ; 5 mS Verzoegerung
          bsr zeit
uspr:     move.w (sp)+,d0                   ; Alter Vergleichswert
          move.b d0,letzt_a2                ; als Byte
          movem.l (sp)+,d0/d2
          move.b d0,letzt_a
          rts

srb:      moveq #4,d1                       ; Taktbit
          move.b d0,letzt_b
          or.b adrsrb,d0                    ; Adressen
          eori.b #%01000000,d0              ; A13 invers
          bra  srxd

src:      moveq #8,d1                       ; Datenschieberegister
          bra  srxd

srx:      move.w (a5),d0                    ; Daten
          moveq #1,d1
decoder:  lsl.w #1,d1
          subq.w #1,2(a5)                   ; Takt
          bne  decoder

; Bedient Schieberegister SRA bis SRC
srxd:     movem.l d2-d3/a0,-(sp)
          lea ciaapb,a0

          move.w rwstat,d3
          move.w d3,d2                      ; 1.Bit
          roxl.b #1,d0
          roxr.b #1,d2
```

ihrem Ermessen, diese Typentabelle beliebig zu erweitern.

Funktionsübersicht

Ein Funktionsaufruf der Juniorsoft-Routinen aus C heraus hat folgendes Grundgerüst:

```
Wert=Junior(Funktionsnummer,
(L:)Parameter,(L:)Parameter,...)
```

Das (L:) zeigt an, daß hier ein Langwort übergeben werden muß.
Der Standardparameter ist vom Datentyp UWORD (siehe auch Listing).
Welche und wieviel Parameter Sie übergeben, hängt von den einzelnen Funktionen ab, die jetzt aufgezählt werden:

Funktion SRX, Nummer 0:
Aufruf:
Void Junior(0,Daten,Sr)
Parameter:
Daten - Das Datenwort für das mit Sr ausgewählte Schieberegister.
Sr - Die Schieberegisternummer. 1 bis 3 entspricht SRA bis SRC.
Zweck:
Transportiert den Parameter "Daten" in das mit SR spezifizierte Ausgangsschieberegister.

Funktion SRD, Nummer 1:
Aufruf:
Byte = Junior(1)
Parameter:
Byte - Die Zustände der Datenleitungen D0-D7.
Zweck:
Gibt als Ergebnis die Zustände der 8 Datenleitungen des Speichers zurück.

Funktion AZ, Nummer 2:
Aufruf:
Void Junior(2,Modus)
Parameter:
Modus - Falls Modus = 0, wird der Adresszähler gelöscht, ansonsten wird er inkrementiert.
Zweck:
Steuerung des Adresszählers.

Funktion SETZEA, Nummer 3:
Aufruf:
Void Junior(3,Ea)
Parameter:
Ea - Bestimmt die Datenrichtung von

oder zum Speicherbaustein. Falls Ea = 0, wird der Speicherbaustein gelesen, bei Ea = 1 ist SRD auf die Datenleitungen geschaltet.

Zweck:

Legt die Datenrichtung von oder zum Speicherbaustein fest.

Ab Funktionsnummer 4 wird zu deren Benutzung die Typentabelle benötigt. Wie diese Tabelle aufgebaut ist und erstellt werden kann, wird später beschrieben.

Funktion Lese, Nummer 4:

Aufruf:

Ergebnis =Junior(4,L:Ss,L:Es,L:Ee)

Parameter:

Ss - Bestimmt die Speicherstartadresse im Rechner, ab der die Daten aus dem Speicherbaustein abgelegt werden.

Es - Die Startadresse im Speicherbaustein.

Ee - Die Endadresse im Speicherbaustein.

Ergebnis:

Enthält die letzte gelesene Adresse (=Ee).

Zweck:

Mit dieser Funktion wird ein beliebiger Bereich des Speicherbausteins in den Speicher des Rechners gelesen.

Funktion Ltest, Nummer 5:

Aufruf:

Ergebnis =Junior(5,L:Es,L:Ee)

Parameter:

Es - Wie bei Funktion 4.

Ee - Wie bei Funktion 4.

Ergebnis:

Falls der Speicherbaustein im spezifizierten Bereich gelöscht ist, erhält man in Ergebnis die letzte Adresse (=Ee). Ist der Speicherbaustein im spezifizierten Bereich nicht gelöscht, so wird die fehlerhafte Adresse des Speicherbausteins mit gesetztem, höchstwertigen Bit zurückgegeben.

Zweck:

Überprüfen des spezifizierten Bereichs im Speicherbaustein auf alle Datenbits = 1.

Funktion Vergl, Nummer 6:

Aufruf:

Ergebnis =Junior(6,L:Ss,L:Es,L:Ee)

Parameter:

Ss,Es,Ee sind definiert wie bei Funktion 4.

Ergebnis ist definiert wie bei Funktion 5, mit der Ausnahme, daß hier auf

```
        move.b d2,(a0)
        or.b d1,d2
        move.b d2,(a0)
        move.w d3,d2                    ; 2.Bit
        roxl.b #1,d0
        roxr.b #1,d2
        move.b d2,(a0)
        or.b d1,d2
        move.b d2,(a0)
        move.w d3,d2                    ; 3.Bit
        roxl.b #1,d0
        roxr.b #1,d2
        move.b d2,(a0)
        or.b d1,d2
        move.b d2,(a0)
        move.w d3,d2                    ; 4.Bit
        roxl.b #1,d0
        roxr.b #1,d2
        move.b d2,(a0)
        or.b d1,d2
        move.b d2,(a0)
        move.w d3,d2                    ; 5.Bit
        roxl.b #1,d0
        roxr.b #1,d2
        move.b d2,(a0)
        or.b d1,d2
        move.b d2,(a0)
        move.w d3,d2                    ; 6.Bit
        roxl.b #1,d0
        roxr.b #1,d2
        move.b d2,(a0)
        or.b d1,d2
        move.b d2,(a0)
        move.w d3,d2                    ; 7.Bit
        roxl.b #1,d0
        roxr.b #1,d2
        move.b d2,(a0)
        or.b d1,d2
        move.b d2,(a0)
        move.w d3,d2                    ; 8.Bit
        roxl.b #1,d0
        roxr.b #1,d2
        move.b d2,(a0)
        or.b d1,d2
        move.b d2,(a0)

        or.b #strobe,(a0)               ; Strobe =1
        and.b #255-strobe,(a0)          ; Strobe = 0
        movem.l (sp)+,d2-d3/a0
        rts

srd:
; Bedient srd
srdd:   movem.l d3-d5/a1-a2,-(sp)       ; Eingangsschiebe
                                          register lesen
        lea ciaapb,a1
        move.b (a1),d3                  ; Strobeimpuls geben
        or.b #strobe,(a1)
        lea ciabpa,a2
        and.b #255-strobe,(a1)          ; Strobe low
        moveq #32,d3                    ; Register vorbesetzen

        moveq #0,d4
        moveq #0,d0

        move.b (a2),d5                  ;1
        roxr.b #1,d5
        move.b d3,(a1)
        addx.w d0,d0
        move.b d4,(a1)

        move.b (a2),d5                  ;2
        roxr.b #1,d5
        move.b d3,(a1)
```

```
          addx.w d0,d0
          move.b d4,(a1)

          move.b (a2),d5                 ;3
          roxr.b #1,d5
          move.b d3,(a1)
          addx.w d0,d0
          move.b d4,(a1)

          move.b (a2),d5                 ;4
          roxr.b #1,d5
          move.b d3,(a1)
          addx.w d0,d0
          move.b d4,(a1)

          move.b (a2),d5                 ;5
          roxr.b #1,d5
          move.b d3,(a1)
          addx.w d0,d0
          move.b d4,(a1)

          move.b (a2),d5                 ;6
          roxr.b #1,d5
          move.b d3,(a1)
          addx.w d0,d0
          move.b d4,(a1)

          move.b (a2),d5                 ;7
          roxr.b #1,d5
          move.b d3,(a1)
          addx.w d0,d0
          move.b d4,(a1)

          move.b (a2),d5                 ;8
          roxr.b #1,d5
          addx.w d0,d0

          not.b d0                       ;Datenwort invertieren
         movem.l (sp)+,d3-d5/a1-a2
          rts


az:       tst.w (a5)                     ; Adresszaehler
                                           inkrement./loeschen
          beq lazd

; Adresszaehler erhoehen
azd:      ori.b #azb,ciaapb
         and.b #255-azb,ciaapb
          rts

; Adresszaehler loeschen
lazd:     move.b letzt_a,d0
          move.w d0,-(sp)
          or.b #lazb,d0
          bsr sra
          move.w (sp)+,d0
          and.b #255-lazb,d0
          bsr sra
          rts

setzea: move.w (a5),d0                   ; Eingabe / Ausgabe
                                           status setzen
          and.w #1,d0

; setzt E/A Status
setzead: move.b d0,ciaapb
          lsl.w #1,d0
          move.w d0,rwstat
          rts

; setzt Adresse in d0 auf den virtuellen Speicher
setzadr: movem.l d1-d4/a0/a1,-(sp)
          move.l d0,d2
```

Gleichheit der Daten geprüft wird.

Zweck:
Vergleicht den spezifizierten Speicherbereich des Speicherbausteins mit dem des Rechners.

Funktion Vtest, Nummer 7:

Aufruf:
Ergebnis =Junior(7,L:Ss,L:Es,L:Ee)

Parameter:
Alle Parameter definiert wie bei Nummer 6, mit der Ausnahme, daß hier geprüft wird, ob ein EPROM, das bereits programmiert ist, erneut mit den Daten ab Ss programmiert werden kann.

Zweck:
Prüfen, ob ein erneutes Programmieren des Speichers möglich ist. Das ist möglich, da einzelne Bits eines EPROMs elektrisch auf Null programmiert werden können, jedoch nicht elektrisch auf 1. So wird hiermit lediglich geprüft, daß die aktuellen Daten kein Bit auf 1 setzen, welches im EPROM schon auf Null gesetzt ist.

Funktion Ipro, Nummer 8:

Aufruf:
Ergebnis=Junior(8,L:Ss,L:Es,L:Ee,Flagge)

Parameter:
Ss,Es,Es und Ergebnis sind definiert wie bei Nummer 6, mit dem Unterschied, daß Ergebnis die erste fehlerhafte Adresse enthält.

Ist Flagge = 0, so werden alle $FF-Bytes bei der Programmierung übergangen, was nur bei EPROMs sinnvoll ist. Ist Flagge jedoch 1, so werden alle Bytes programmiert.

Zweck:
Programmiert einen Speicherbaustein in dem spezifizierten Bereich mit dem Inhalt des spezifizierten Speicherbereich des Rechners. Ausgehend von dem standardisierten "intelligenten" Programmieralgorithmus, wie er in verschiedenen Datenblättern vorgeschlagen wird, können alle anderen Algorithmen durch Parametervariation in der Konfigurationstabelle von diesem abgeleitet werden.

Funktion Tstest, Nummer 9:

Aufruf:
Ergebnis = Junior(9)

Parameter:
keineErgebnis ist 0, 1 oder 2, falls der Speicherbaustein keine gültige Rückmeldung gibt, also nicht oder nicht

richtig eingesetzt ist. Ist Ergebnis = 0, dann ist höchstwahrscheinlich ein funktionsfähiger Speicherbaustein eingesetzt.
Zweck:
Prüfen, ob ein Speicher eingesetzt ist.

Funktion Waltp, Nummer 10:
Aufruf:
Void Junior(10,L:Typ)
Parameter:
Typ ist die Adresse, an der die dem Typ entsprechende, 82 Bytes lange Konfigurationstabelle stehen muß.
Zweck:
Einstellen des Speichertyps.

Funktion Init, Nummer 11:
Aufruf:
Void Junior(11)
Parameter:
keine
Zweck:
Initialisiert den Juniorprommer anfangs für den weiteren Betrieb.

Die Typenkonfigurationstabelle

Wie Sie jetzt wissen, können Sie fast jeden 28- oder 24-poligen Speicherbaustein mit Hilfe dieser Tabelle in den Juniorprommer einbinden. In dieser Tabelle sind alle möglichen Daten von der Typenbezeichnung bis zur Programmierspannung enthalten. Listing 2 zeigt diese Typentabelle bereits erstellt für die EPROMs 2716 bis 27011. Ein kompletter Eintrag, welcher jeweils einen Speichertyp repräsentiert, ist genau 82 Bytes lang, muß an einer geraden Adresse beginnen und ist so aufgebaut:

1. Eintrag: Zkette

Offset: 0
Datentyp: Zeichenkette mit konstant 30 Zeichen.

Die Zeichenkette dient lediglich der Benutzerorientierung und beschreibt den Speichertyp in Kurzform.

2. Eintrag: Kapazi

Offset: 30
Datentyp: Langwort.

```
         move.l typzeig,a1
wdh:     cmp.l istadr,d2
         beq stimmt
         bhi gr
         clr.l istadr
         bsr lazd
         bra nerh
gr:      addq.l #1,istadr              ; Adresse erhoehen
         bsr azd                       ; Adresszaehler auch
nerh:    move.l istadr,d1
         lsr.l #8,d1
         lsr.l #2,d1
         cmp.l letzta10,d1             ; gleiche Adresse ??
         beq wdh                       ; Ja,dann unnoetig
         tst.l seite(a1)               ; gebankter Speicher
         beq kbea
         move.l istadr,d3
         divu seite+2(a1),d3           ; Seite errechnen
         cmp.w aseite,d3               ; gleich der Bisherigen ?
         beq kbea
         move.w d3,aseite
         bsr seitesz
kbea:    move.l d1,letzta10            ; jetzt gleich
         move.l a1,a0
         add.w #m_a10,a0
         moveq #6,d0                   ; maximal 7 Adresspins
         clr.w d3
         clr.w d4
setzax:  lsr.w #1,d1                   ; Adresse
         bcc axnull
         or.b (a0),d3                  ; SRA
         or.b 1(a0),d4                 ; SRB
axnull:  addq.l #2,a0
         dbra d0,setzax
         move.b d3,adrsra              ; Oder-Adressen
         move.b d4,adrsrb
         move.b letzt_a,d0             ; Adresse aktualisieren
         movem.l a0-a1,-(sp)
         bsr sra
         move.b letzt_b,d0
         bsr srb
         movem.l (sp)+,a0-a1
         bra wdh
stimmt:  movem.l (sp)+,d1-d4/a0/a1
         rts

seitesz: movem.l d0-d3/a0-a1,-(sp)
         move.l typzeig,a1
         move.w aseite,d0              ; Aktuelle Speicherbank
         bsr src                       ; Auf den Datenbus
         moveq #1,d0
         bsr setzead                   ; Schreiben
         move.b letzt_a,d2             ; SRA
         move.b letzt_b,d3             ; SRB
         move.b d2,d0
         andi.b #%11100111,d0          ; evtl. Progammiersp.
                                         ausschalten
         bsr sra
         move.w #100,d0
         bsr zeit                      ; 10 mS Warten
         move.b odis+1(a1),d0
         bsr srb
         move.b page_wr+1(a1),d0
         bsr srb
         move.b odis+1(a1),d0
         bsr srb
         move.b d2,d0
         bsr sra
         move.b d3,d0
         bsr srb
         move.w #20,d0
         bsr zeit                      ; 2mS Warten
         moveq #0,d0
         bsr setzead
```

"Kapazi" muß die Kapazität des Speichertyps in Bytes enthalten.

3. Eintrag: Seite

```
Offset: 34
Datentyp: Langwort.
```

"Seite" muß die Seitengröße eines gebankten Speichers in Byte enthalten. Falls der zu spezifizierende Speicher nicht gebankt ist, muß "Seite" auf Null gesetzt sein.

4. Eintrag: Alg

```
Offset: 38
Datentyp: Wort.
```

Dient lediglich der Benutzerorientierung und kann einen Wert enthalten, der Aufschluß über die möglichen Programmieralgorithmen gibt.

5. Eintrag: I_Zeit

```
Offset: 40
Datentyp: Wort.
```

Bestimmt die Programmierdauer des Initialprogrammierimpulses in 0.1ms-Schritten.

6. Eintrag: M_Zeit

```
Offset: 42
Datentyp: Wort.
```

Bestimmt die Zeitdauer, welche, mit der Anzahl der Initialimpulse multipliziert, die Nachprogrammierdauer in 0.1ms-Schritten ergibt.
Ist M_Zeit = 0, so wird das Nachprogrammieren übergangen.

7. Eintrag: Trials

```
Offset: 44
Datentyp: Wort.
```

Legt die Anzahl der Initialprogrammierversuche fest, nach denen die Programmierung abgebrochen wird.

8. Eintrag: Standby

```
Offset: 46
Datentyp: 2 Bytes.
```

```
                movem.l (sp)+,d0-d3/a0-a1
                 rts

        sraus:  move.b d2,d0
                bsr sra
                move.b d3,d0
                bra srb

        ; Legt Kontrollzustaende bei A0 an den virtuellen Speicher
        setzst: move.l a0,-(sp)
                move.b (a0)+,d0
                cmp.b letzt_a,d0
                beq ubf1
                bsr sra                         ; Schieberegister a
        ubf1:   move.b (a0)+,d0
                cmp.b letzt_b,d0
                beq ubf2
                bsr srb
        ubf2:   move.l (sp)+,a0
                rts

        ; Zeitverzoegerung
        zeit:   tst.w d0
                beq nullzt
                mulu #100,d0                    ; in 0.1ms Schritten
                sub.l #70,d0
                movem.l d1-d7/a0-a6,-(sp)
                movem.l save,d1-d7/a0-a6        ; Environment
                move.l d0,-(sp)                 ; An Funktion uebergeben
                jsr _WaitForTimer
                addq.l #4,sp
                movem.l (sp)+,d1-d7/a0-a6
        nullzt: rts


        ; Aktiviert den Speicher und loescht die Adresse
        einsch: move.b #$ff,ciaapbd
                clr.l adrsra
                clr.l letzta10
                clr.w letzt_a
                clr.w letzt_a2
                clr.w letzt_b
                clr.w aseite
                moveq #0,d0                     ; Ausgangsschiebe
                                                  register aus
                bsr setzead
                move.l typzeig,a0               ; Aktuelles EPROM
                add.w #odis,a0                  ; Betriebszustand
                bsr setzst
                bsr seitesz                     ; Falls noetig, Seite
                                                  loeschen
                clr.l d0                        ; Adresse zuruecksetzen
                bra setzadr

        ; Inaktiviert den Speicher und loescht die Adresse
        aussch: moveq #0,d0
                bsr setzead
                move.l typzeig,a0
                add.w #standby,a0
                bsr setzst
                clr.l d0
                bra setzadr

        ; Liest den Speicher in den des Rechners
        ; Parameter : 0/Adresse.l,4/Epromstart,8/Epromende
        lese:   move.l (a5),a2                  ; Speicheradresse
                move.l 4(a5),d3                 ; Epromstart
                move.l 8(a5),d4                 ; Epromende
                bsr einsch                      ; Speicher aktivieren
                move.l typzeig,a0
                add.w #read,a0
                bsr setzst
        lesesc: move.l d3,d0
                bsr setzadr                     ; Adresse setzen
```

Enthält die Kontrollbytes für den spezifischen Betriebszustand "Standby" des Speichers. Dieser Betriebszustand ist, wie alle anderen auch, in den Datenblättern der Hersteller beschrieben. Das erste Kontrollbyte korrespondiert direkt mit SRA, das zweite direkt mit SRB.

9. Eintrag: Read

```
Offset: 48
Datentyp: 2 Bytes.
```

Wie zuvor, jedoch wird der Betriebszustand "Read" beschrieben.

10. Eintrag: Odis

```
Offset: 50
Datentyp: 2 Bytes.
```

Bestimmt den Betriebszustand "Output Disable".

11. Eintrag: Program

```
Offset: 52
Datentyp: 2 Bytes.
```

Bestimmt den Betriebszustand "Program".

12. Eintrag: Pr_ver

```
Offset: 54
Datentyp: 2 Bytes.
```

Bestimmt den Betriebszustand "Program Verify".

13. Eintrag: Pr_inh

```
Offset: 56
Datentyp: 2 Bytes.
```

Bestimmt den Bestriebszustand "Program Inhibit".

14. Eintrag: Page_wr

```
Offset: 58
Datentyp: 2 Bytes.
```

Bestimmt den Betriebszustand "Page Write".

15. Eintrag: Manufac

```
Offset: 60
Datentyp: Byte.
```

```
        bsr srd                         ; Datenwort reinschieben
        move.b d0,(a2)+
        addq.l #1,d3
        cmp.l d3,d4
        bcc lesesc
        bsr aussch
        move.l d3,d0
        subq.l #1,d0
        rts

; Macht Leertest des Speichers
; Parameter : 0/Epromstart,4/Epromende
ltest:  move.l 0(a5),d3                 ; Epromstart
        move.l 4(a5),d4                 ; Epromende
        bsr einsch                      ; Speicher aktivieren
        move.l typzeig,a0
        add.w #read,a0
        bsr setzst
ltsc:   move.l d3,d0
        bsr setzadr                     ; Adresse setzen
        bsr srd                         ; Datenwort reinschieben
        cmp.b #$ff,d0                   ; Leertest
        bne ltf
        addq.l #1,d3
        cmp.l d3,d4
        bcc ltsc
        bsr aussch
        move.l d3,d0
        subq.l #1,d0
        rts
ltf:    bsr aussch
        move.l d3,d0
        subq.l #1,d0
        bset #31,d0
        rts

; Vergleicht den Speicher mit dem des Rechners
; Parameter : 0/Adresse.l,4/Epromstart,8/Epromende
vergl:  move.l (a5),a2                  ; Speicheradresse
        move.l 4(a5),d3                 ; Epromstart
        move.l 8(a5),d4                 ; Epromende
        bsr einsch                      ; Speicher aktivieren
        move.l typzeig,a0
        add.w #read,a0
        bsr setzst
vgsc:   move.l d3,d0
        bsr setzadr                     ; Adresse setzen
        bsr srd                         ; Datenwort reinschieben
        cmp.b (a2)+,d0
        bne vgf
        addq.l #1,d3
        cmp.l d3,d4
        bcc vgsc
        bsr aussch
        move.l d3,d0
        subq.l #1,d0
        rts
vgf:    bsr aussch
        move.l d3,d0
        subq.l #1,d0
        bset #31,d0
        rts

; Fuerht einen Vertraeglichkeitstest der Daten aus
; Parameter : 0/Adresse.l,4/Epromstart,8/Epromende
vtest:  move.l (a5),a2                  ; Speicheradresse
        move.l 4(a5),d3                 ; Epromstart
        move.l 8(a5),d4                 ; Epromende
        bsr einsch                      ; Speicher aktivieren
        move.l typzeig,a0
        add.w #read,a0
        bsr setzst
vtsc:   move.l d3,d0
        bsr setzadr                     ; Adresse setzen
```

```
        bsr srd                     ; Datenwort reinschieben
        not.b d0                    ; invertieren
        and.b (a2)+,d0
        bne vtf
        addq.l #1,d3
        cmp.l d3,d4
        bcc vtsc
        bsr aussch
        move.l d3,d0
        subq.l #1,d0
        rts
vtf:    bsr aussch
        move.l d3,d0
        subq.l #1,d0
        bset #31,d0
        rts

; Programmierroutine
ipro:   move.l (a5),a3              ; Speicheradresse
        move.l 4(a5),d3             ; Epromstartadresse
        move.l 8(a5),d4             ; Epromendadresse
        move.w 12(a5),d5            ; $ff-Flagge
        bsr einsch                  ; Speicher aktivieren
        move.l typzeig,a4
progby:
        move.l d3,d0
        bsr setzadr                 ; Adresse
        move.b (a3),d0
        tst.w d5                    ; $FF Ueberspringen ??

        bne uff
        cmpi.b #$ff,d0
        beq ladr
uff:    bsr src                     ; An das Eprom
        clr.w d2                    ; Versuchszaehler
ippuls:
        lea pr_inh(a4),a0
        bsr setzst
        moveq #1,d0
        bsr setzead
        lea program(a4),a0
        bsr setzst                  ; Programmierpuls
        move.w i_zeit(a4),d0
        bsr zeit                    ; Verzoegerung
        lea pr_inh(a4),a0
        bsr setzst                  ; aus
        addq.w #1,d2                ; Versuchszaehler
        cmp.w trials(a4),d2         ; Vergleichen
        beq lpuls
        moveq #0,d0
        bsr setzead
        lea pr_ver(a4),a0           ; Daten Vergleichen
        bsr setzst
        bsr srd
        cmp.b (a3),d0
        bne ippuls                  ; Nochmal
lpuls:  tst.w m_zeit(a4)            ; Multiplikationsfaktor
        beq nichts
        lea pr_inh(a4),a0
        bsr setzst
        moveq #1,d0
        bsr setzead
        lea program(a4),a0          ; 2. Programmierpuls
        bsr setzst
        move.w d2,d0
        mulu m_zeit(a4),d0          ; Multiplikationsfaktor
        bsr zeit
        lea pr_inh(a4),a0
        bsr setzst
nichts: cmp.w trials(a4),d2         ; Maximalzeit ?
        bne ladr
        moveq #0,d0
        bsr setzead                 ; Daten einlesen
        lea pr_ver(a4),a0
```

Hat hier keine Bedeutung, würde jedoch den Herstellercode des Speichers angeben.

16. Eintrag: Devcode

```
Offset: 61
Datentyp: Byte.
```

Wie zuvor, jedoch würde dieses Byte den Typencode angeben.

17. Eintrag: Tsmask

```
Offset: 62
Datentyp: Byte.
```

Hat hier keine Bedeutung.

18. Eintrag: Frei3

```
Offset: 63
Datentyp: Byte.
```

Unbenutzter Eintrag.

19. Eintrag: Frei1

```
Offset: 64
Datentyp: Wort.
```

Unbenutzter Eintrag.

20. Eintrag: Frei2

```
Offset: 66
Datentyp: Wort.
```

Unbenutzter Eintrag.

21. Eintrag: M_a10

```
Offset: 68
Datentyp: 2 Byte.
```

Bestimmt den Pin, der bei dem zu beschreibenden Speicher mit A10 benannt ist. Das erste Byte korrespondiert hier wieder direkt mit SRA, das zweite Byte direkt mit SRB. Da diese Daten mit den Betriebszuständen oderverknüpft werden, müssen die Betriebszustande für alle spezifizierten Adresspins Null sein.

22. Eintrag: M_a11

```
Offset: 70
Datentyp: 2 Byte.
```

Wie zuvor, jedoch für A11.
Die Tabelle wird chronologisch bis zum 27. Eintrag, der A16 beschreibt, fortgeführt.
In Listing 2 finden Sie Beispiele mit den Eprom-Typen 2716 bis 27011.

Handhabung der Konfigurationstabelle in der Praxis

Falls Sie einen neuen Typ implementieren wollen, sollten Sie zuerst prüfen, ob dieser überhaupt geeignet ist. Die Steuerleitungen und restlichen Adressleitungen sollten von SRA oder SRB abgedeckt werden, und die Adress-, Daten-, Versorgungs- und Programmierspannungsleitungen, welche bereits fest verdrahtet sind, sollten an den festgelegten Pins liegen. Zu beachten ist, daß Speicher mit bis zu 17 Adressleitungen möglich sind, wie z.B. die Rombausteine, die neuerdings in ST2/ST4-Modellen Verwendung finden. Nun füllen Sie am besten die Konfigurationstabelle mit Hilfe des Datenblatts des Speichertyps aus. Einträge in der Konfigurationstabelle, die von Juniorsoft nicht explizit genutzt werden, können Sie von Ihrem eigenen Programm aus belegen. Zuletzt sollten Sie nicht vergessen, die LED auch in den verschiedenen Kontrollzuständen des EPROMs zu berücksichtigen.
Die Konfigurationstabelle hat, wie schon gesagt, 82 Bytes pro Eintrag (Typ). Sie sollten immer darauf achten, daß genug freier Speicher für die Konfigurationstabelle vorhanden ist. Wo Sie die Konfigurationstabelle im Arbeitsspeicher ablegen, muß immer den Gegebenheiten angepaßt werden. In C bietet sich z.B. die Möglichkeit an, diese in einer eigenen Struktur abzulegen, die dann mit den entsprechenden Werten zu initialisieren ist.

Die Juniorsoftfunktionen in der Praxis

Eine Beispielanwendung der Juniorsoft-Funktionen ist bereits in Listing 3 gezeigt.

```
         bsr setzst
         bsr srd
         cmp.b (a3),d0
         bne pfeh
ladr:
         addq.l #1,d3                  ; Epromadresse
         addq.l #1,a3                  ; Speicheradresse
         cmp.l d3,d4
         bcc progby                    ; Bis zur Epromend
                                         adresse
         bsr aussch                    ; Speicher abschalten
         move.l d3,d0
         subq.l #1,d0
         rts
pfeh:    bsr aussch
         move.l d3,d0
         subq.l #1,d0
         bset #31,d0
         rts

waltp:   move.l (a5),typzeig        ; Typenzeiger setzen
         bsr einsch
         bra aussch

; Prueft die Treiberfaehigkeit des eingesetzten Speichers

tstest: bsr einsch                     ; Aktivieren
         move.l typzeig,a4
         clr.l d0
         bsr setzadr
         lea read(a4),a0               ; Lesestatus
         bsr tstbd
         move.b d3,d5
         move.b d4,d6
         lea odis(a4),a0
         bsr tstbd
         tst.b d3
         beq  ed
         tst.b d4
         beq  ed
         tst.b d5
         beq  ed
         tst.b d6
         beq  ed
         bsr aussch
         moveq #2,d0
         rts

ed:      cmp.b d5,d6
         bne  fehl
         cmp.b d3,d4
         beq  fehl
         bsr aussch
         moveq #0,d0
         rts
fehl:
         bsr aussch
         moveq #1,d0
         rts


tstbd:   move.l a0,-(sp)
         bclr #1,(a0)
         bsr setzst
         bsr srd
         btst #7,d0
         seq d3
         move.l (sp)+,a0
         bset #1,(a0)
         bsr setzst
         bsr srd
         btst #7,d0
         seq d4
```

```
            rts


init:   move.b #$ff,ciaapbd
        clr.l letztal0
        clr.l adrsra
        clr.l letzt_a
        clr.w letzt_b
        lea inaktiv,a0
        bsr setzst
        clr.l d0
        bra setzadr

    dseg

inaktiv: dc.b 0
         dc.b 0

stopikb: dc.b $13
startikb: dc.b $11

; lokale Variablen

ergebnis: dc.l $23443333
alt_ssp: dc.l $23334440
alt_sr:  dc.l $0
rwstat:  dc.w $2323
aseite:  dc.w 0

save:   dc.l 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
istadr: dc.l $44444444
typzeig: dc.l $44444444
adrsra: dc.w 0
adrsrb: dc.w 0
letzt_a: dc.w 0
letzt_a2: dc.w 0
letzt_b: dc.w 0

letztal0: dc.l 1
```

Listing 1: Der Assemblerquellcode "junior.asm" der Juniorsoft-Funktionen in seiner Version für Aztec Assembler.

Wie schon gesagt, dienen die Funktionen 0-3 zur Ansteuerung des Juniorprommers auf niedrigster Ebene. Die Funktionen 4-11 bauen auf diesen auf und ermöglichen die Ansteuerung des Juniorprommers entsprechend einem Massenspeicher, benötigen zum Betrieb jedoch die Typentabelle. Es muß immer vor dem Aufruf dieser Funktionen ein gültiger Typ eingestellt werden, da es sonst zu Abstürzen kommen kann. Die meisten Funktionen wie z.B. Lesen und Programmieren benötigen einen Datenspeicher im Rechner. Dieser kann z.B. in einem Array gehalten werden.

Alle Adressparameter der Funktionen werden nicht auf Gültigkeit geprüft, denn das liegt im Aufgabenbereich des Steuerprogramms. Außerdem sollten Adressvariablen ausschließlich vom Typ ULONG sein.

In Listing 4 ist ein Programm abgedruckt, das anhand dem Kopieren eines EPROMs vom Typ 27128 fast alle Funktionen benutzt und erläutert.

Und zum Schluß

Falls Sie wenig Interesse daran haben, ihr "eigenes" EPROM zu programmieren, besteht auch noch die Möglichkeit, daß sie eine fertige Software über Merlin Computer GmbH beziehen. Auch die Hardware ist entweder als Platine oder als Fertiggerät erhältlich.

```
; Typentabelle der Eproms 2716 bis 27011
; (c) Merlin Computer   KICKSTART
; erstellt am 20.3.88
; R. Hofmann
;
; Aztec C 3.6
; as typtab -N -C -D

  dseg
  public _TypTab    ; Schnittstelle zur Aussenwelt

_TypTab:              ; Aus C liefert Typtab die
                        Tabelle
                      ; des Eprom 2716. Wird der
                        Pointer
                      ; *EpromTyp wie im Beispiel
                        verwendet
                      ; kann mit EpromTyp+=5 der
                        fuenfte
                      ; Index erreicht werden ->
                        27128 21V

; Typentabelle zu junior.asm
; Typen 2716 bis 27011

; Typ 2716, Index 0

  dc.b "2716  EPROM  2Kb 25V          "
                    ; Beschreibung
  dc.l 2048       ; Kapazitaet in Bytes
  dc.l 0          ; Groesse einer Seite
  dc.w 0          ; Programmieralgoritmus
  dc.w 500        ; i_zeit
  dc.w 00         ; m_zeit
  dc.w 2          ; trials
;      AAAAAAAA  BBBBBBBB
;      76543210  76543210

  dc.b %00001000,%01000011      ; Standby
  dc.b %00001010,%01000000      ; Read
  dc.b %00001010,%01000010      ; Odis
  dc.b %10001010,%01000011      ; Program
  dc.b %10001010,%01000000      ; Pr_ver
  dc.b %10001010,%01000010      ; Pr_inh
  dc.b %00000000,%00000000      ; Page_wr
  dc.b 0                        ; Manufacturer Code
  dc.b 0                        ; Device Code
  dc.b %00000000                ; Tsmask
  dc.b 0                        ; Frei 3
  dc.b %00000000,%00000000      ; Frei 1
  dc.b %00000000,%00000000      ; Frei 2

  dc.b %00000000,%00001000      ; Tabelle A10
  dc.b %00000000,%00000000      ; A11
  dc.b %00000000,%00000000      ; A12
  dc.b %00000000,%00000000      ; A13
  dc.b %00000000,%00000000      ; A14
  dc.b %00000000,%00000000      ; A15
  dc.b %00000000,%00000000      ; A16

; Typ 2732, Index 1

  dc.b "2732  EPROM  4Kb 25V          "
                                ; Beschreibung
```

```
    dc.l 4096      ; Kapazitaet in Bytes
    dc.l 0         ; Groesse einer Seite
    dc.w 0        ; Programmieralgoritmus
    dc.w 500       ; i_zeit
    dc.w 0         ; m_zeit
    dc.w 2         ; trials
;       AAAAAAAA  BBBBBBBB
;       76543210  76543210

    dc.b %00000000,%01000011       ; Standby
    dc.b %00000010,%01000000       ; Read
    dc.b %00000010,%01000010       ; Odis
    dc.b %10000010,%01000110       ; Program
    dc.b %10000010,%01000000       ; Pr_ver
    dc.b %10000010,%01000010       ; Pr_inh
    dc.b %00000000,%00000000       ; Page_wr
    dc.b 0                         ; Manufacturer Code
    dc.b 0                         ; Device Code
    dc.b %00000000                 ; Tsmask
    dc.b 0                         ; Frei 3
    dc.b %00000000,%00000000       ; Frei 1
    dc.b %00000000,%00000000       ; Frei 2

    dc.b %00000000,%00001000       ; Tabelle A10
    dc.b %00000000,%00010000       ; A11
    dc.b %00000000,%00000000       ; A12
    dc.b %00000000,%00000000       ; A13
    dc.b %00000000,%00000000       ; A14
    dc.b %00000000,%00000000       ; A15
    dc.b %00000000,%00000000       ; A16

; Typ 2732 A, Index 2

    dc.b "2732A EPROM  4Kb 21V          "       ; Beschreibung
    dc.l 4096      ; Kapazitaet in Bytes
    dc.l 0         ; Groesse einer Seite
    dc.w 0        ; Programmieralgoritmus
    dc.w 500       ; i_zeit
    dc.w 0         ; m_zeit
    dc.w 2         ; trials
;       AAAAAAAA  BBBBBBBB
;       76543210  76543210

    dc.b %00000000,%01000011       ; Standby
    dc.b %00000010,%01000000       ; Read
    dc.b %00000010,%01000010       ; Odis
    dc.b %01000010,%01000110       ; Program
    dc.b %01000010,%01000000       ; Pr_ver
    dc.b %01000010,%01000010       ; Pr_inh
    dc.b %00000000,%00000000       ; Page_wr
    dc.b 0                         ; Manufacturer Code
    dc.b 0                         ; Device Code
    dc.b %00000000                 ; Tsmask
    dc.b 0                         ; Frei 3
    dc.b %00000000,%00000000       ; Frei 1
    dc.b %00000000,%00000000       ; Frei 2

    dc.b %00000000,%00001000       ; Tabelle A10
    dc.b %00000000,%00010000       ; A11
    dc.b %00000000,%00000000       ; A12
    dc.b %00000000,%00000000       ; A13
    dc.b %00000000,%00000000       ; A14
    dc.b %00000000,%00000000       ; A15
    dc.b %00000000,%00000000       ; A16

; Typ 2764, Index 3

    dc.b "2764  EPROM  8Kb 21V          "       ; Beschreibung
    dc.l 8192      ; Kapazitaet in Bytes
    dc.l 0         ; Groesse einer Seite
    dc.w 0        ; Programmieralgoritmus
    dc.w 10        ; i_zeit
    dc.w 40        ; m_zeit
    dc.w 15        ; trials
;       AAAAAAAA  BBBBBBBB
;       76543210  76543210

    dc.b %00010010,%10000011       ; Standby
    dc.b %00010010,%10000000       ; Read
    dc.b %00010010,%10000001       ; Odis
    dc.b %01010010,%00000010       ; Program
    dc.b %01010010,%10000000       ; Pr_ver
    dc.b %01010010,%10000010       ; Pr_inh
    dc.b %00000000,%00000000       ; Page_wr
    dc.b 0                         ; Manufacturer Code
    dc.b 0                         ; Device Code
    dc.b %00000000                 ; Tsmask
    dc.b 0                         ; Frei 3
    dc.b %00000000,%00000000       ; Frei 1
    dc.b %00000000,%00000000       ; Frei 2

    dc.b %00000000,%00001000       ; Tabelle A10
    dc.b %00000000,%00010000       ; A11
    dc.b %00000000,%00100000       ; A12
    dc.b %00000000,%00000000       ; A13
    dc.b %00000000,%00000000       ; A14
    dc.b %00000000,%00000000       ; A15
    dc.b %00000000,%00000000       ; A16

; Typ 2764A, Index 4

    dc.b "2764A EPROM  8Kb 12.5V        "       ; Beschreibung
    dc.l 8192      ; Kapazitaet in Bytes
    dc.l 0         ; Groesse einer Seite
    dc.w 0        ; Programmieralgoritmus
    dc.w 10        ; i_zeit
    dc.w 40        ; m_zeit
    dc.w 15        ; trials
;       AAAAAAAA  BBBBBBBB
;       76543210  76543210

    dc.b %00010010,%10000011       ; Standby
    dc.b %00010010,%10000000       ; Read
    dc.b %00010010,%10000001       ; Odis
    dc.b %00110010,%00000010       ; Program
    dc.b %00110010,%10000000       ; Pr_ver
    dc.b %00110010,%10000010       ; Pr_inh
    dc.b %00000000,%00000000       ; Page_wr
    dc.b 0                         ; Manufacturer Code
    dc.b 0                         ; Device Code
    dc.b %00000000                 ; Tsmask
    dc.b 0                         ; Frei 3
    dc.b %00000000,%00000000       ; Frei 1
    dc.b %00000000,%00000000       ; Frei 2

    dc.b %00000000,%00001000       ; Tabelle A10
    dc.b %00000000,%00010000       ; A11
    dc.b %00000000,%00100000       ; A12
    dc.b %00000000,%00000000       ; A13
    dc.b %00000000,%00000000       ; A14
    dc.b %00000000,%00000000       ; A15
    dc.b %00000000,%00000000       ; A16

; Typ 27128, Index 5

    dc.b "27128 EPROM 16Kb 21V          "       ; Beschreibung
    dc.l 16384     ; Kapazitaet in Bytes
    dc.l 0         ; Groesse einer Seite
    dc.w 0        ; Programmieralgoritmus
    dc.w 10        ; i_zeit
    dc.w 40        ; m_zeit
    dc.w 15        ; trials
;       AAAAAAAA  BBBBBBBB
;       76543210  76543210

    dc.b %00010000,%10000011       ; Standby
    dc.b %00010010,%10000000       ; Read
    dc.b %00010010,%10000010       ; Odis
    dc.b %01010010,%00000010       ; Program
    dc.b %01010010,%10000000       ; Pr_ver
    dc.b %01010010,%10000010       ; Pr_inh
    dc.b %00000000,%00000000       ; Page_wr
    dc.b 0                         ; Manufacturer Code
    dc.b 0                         ; Device Code
    dc.b %00000000                 ; Tsmask
    dc.b 0                         ; Frei 3
    dc.b %00000000,%00000000       ; Frei 1
    dc.b %00000000,%00000000       ; Frei 2

    dc.b %00000000,%00001000       ; Tabelle A10
    dc.b %00000000,%00010000       ; A11
```

```
    dc.b %00000000,%00100000        ; A12
    dc.b %00000000,%01000000        ; A13
    dc.b %00000000,%00000000        ; A14
    dc.b %00000000,%00000000        ; A15
    dc.b %00000000,%00000000        ; A16

; Typ 27128A, Index 6

    dc.b "27128A EPROM 16Kb 12.5V        " ; Beschreibung
    dc.l 16384    ; Kapazitaet in Bytes
    dc.l 0        ; Groesse einer Seite
    dc.w 0        ; Programmieralgoritmus
    dc.w 10       ; i_zeit
    dc.w 40       ; m_zeit
    dc.w 15       ; trials
;        AAAAAAAA  BBBBBBBB
;        76543210  76543210

    dc.b %00010010,%10000011        ; Standby
    dc.b %00010010,%10000000        ; Read
    dc.b %00010010,%10000001        ; Odis
    dc.b %00110010,%00000010        ; Program
    dc.b %00110010,%10000000        ; Pr_ver
    dc.b %00110010,%10000010        ; Pr_inh
    dc.b %00000000,%00000000        ; Page_wr
    dc.b 0                          ; Manufacturer Code
    dc.b 0                          ; Device Code
    dc.b %00000000                  ; Tsmask
    dc.b 0                          ; Frei 3
    dc.b %00000000,%00000000        ; Frei 1
    dc.b %00000000,%00000000        ; Frei 2

    dc.b %00000000,%00001000        ; Tabelle A10
    dc.b %00000000,%00010000        ; A11
    dc.b %00000000,%00100000        ; A12
    dc.b %00000000,%01000000        ; A13
    dc.b %00000000,%00000000        ; A14
    dc.b %00000000,%00000000        ; A15
    dc.b %00000000,%00000000        ; A16

; Typ 27256I, Index 7

    dc.b "27256I EPROM 16Kb 12.5V        " ; Beschreibung
    dc.l 32768    ; Kapazitaet in Bytes
    dc.l 0        ; Groesse einer Seite
    dc.w 0        ; Programmieralgoritmus
    dc.w 10       ; i_zeit
    dc.w 30       ; m_zeit
    dc.w 25       ; trials
;        AAAAAAAA  BBBBBBBB
;        76543210  76543210

    dc.b %00010000,%00000011        ; Standby
    dc.b %00010010,%00000000        ; Read
    dc.b %00010010,%00000010        ; Odis
    dc.b %00110010,%00000010        ; Program
    dc.b %00110010,%00000001        ; Pr_ver
    dc.b %00110010,%00000011        ; Pr_inh
    dc.b %00000000,%00000000        ; Page_wr
    dc.b 0                          ; Manufacturer Code
    dc.b 0                          ; Device Code
    dc.b %00000000                  ; Tsmask
    dc.b 0                          ; Frei 3
    dc.b %00000000,%00000000        ; Frei 1
    dc.b %00000000,%00000000        ; Frei 2

    dc.b %00000000,%00001000        ; Tabelle A10
    dc.b %00000000,%00010000        ; A11
    dc.b %00000000,%00100000        ; A12
    dc.b %00000000,%01000000        ; A13
    dc.b %00000000,%10000000        ; A14
    dc.b %00000000,%00000000        ; A15
    dc.b %00000000,%00000000        ; A16

; Typ 27256, Index 8

    dc.b "27256 EPROM 32Kb 21V           " ; Beschreibung
    dc.l 32768    ; Kapazitaet in Bytes
    dc.l 0        ; Groesse einer Seite
    dc.w 0        ; Programmieralgoritmus
    dc.w 10       ; i_zeit
    dc.w 30       ; m_zeit
    dc.w 25       ; trials
;        AAAAAAAA  BBBBBBBB
;        76543210  76543210

    dc.b %00010000,%00000011        ; Standby
    dc.b %00010010,%00000000        ; Read
    dc.b %00010010,%00000010        ; Odis
    dc.b %01010010,%00000010        ; Program
    dc.b %01010010,%00000000        ; Pr_ver
    dc.b %01010010,%00000011        ; Pr_inh
    dc.b %00000000,%00000000        ; Page_wr
    dc.b 0                          ; Manufacturer Code
    dc.b 0                          ; Device Code
    dc.b %00000000                  ; Tsmask
    dc.b 0                          ; Frei 3
    dc.b %00000000,%00000000        ; Frei 1
    dc.b %00000000,%00000000        ; Frei 2

    dc.b %00000000,%00001000        ; Tabelle A10
    dc.b %00000000,%00010000        ; A11
    dc.b %00000000,%00100000        ; A12
    dc.b %00000000,%01000000        ; A13
    dc.b %00000000,%10000000        ; A14
    dc.b %00000000,%00000000        ; A15
    dc.b %00000000,%00000000        ; A16

; Typ 27512, Index 9

    dc.b "27512 EPROM 64Kb 12.5V         "
                                    ; Beschreibung
    dc.l 65536    ; Kapazitaet in Bytes
    dc.l 0        ; Groesse einer Seite
    dc.w 0        ; Programmieralgoritmus
    dc.w 10       ; i_zeit
    dc.w 30       ; m_zeit
    dc.w 25       ; trials
;        AAAAAAAA  BBBBBBBB
;        76543210  76543210

    dc.b %00000000,%00000011        ; Standby
    dc.b %00000010,%00000000        ; Read
    dc.b %00000010,%00000010        ; Odis
    dc.b %00100010,%00000110        ; Program
    dc.b %00100010,%00000000        ; Pr_ver
    dc.b %00100010,%00000001        ; Pr_inh
    dc.b %00000000,%00000000        ; Page_wr
    dc.b 0                          ; Manufacturer Code
    dc.b 0                          ; Device Code
    dc.b %00000000                  ; Tsmask
    dc.b 0                          ; Frei 3
    dc.b %00000000,%00000000        ; Frei 1
    dc.b %00000000,%00000000        ; Frei 2

    dc.b %00000000,%00001000        ; Tabelle A10
    dc.b %00000000,%00010000        ; A11
    dc.b %00000000,%00100000        ; A12
    dc.b %00000000,%01000000        ; A13
    dc.b %00000000,%10000000        ; A14
    dc.b %00000001,%00000000        ; A15
    dc.b %00000000,%00000000        ; A16

; Typ 27513, Index 10

    dc.b "27513 EPROM 4*16Kb*8 12.5V     "
                                    ; Beschreibung
    dc.l 65536    ; Kapazitaet in Bytes
    dc.l 16384    ; Groesse einer Seite
    dc.w 0        ; Programmieralgoritmus
    dc.w 10       ; i_zeit
    dc.w 30       ; m_zeit
    dc.w 25       ; trials
;        AAAAAAAA  BBBBBBBB
;        76543210  76543210

    dc.b %00000001,%10000011        ; Standby
    dc.b %00000011,%10000000        ; Read
    dc.b %00000011,%10000010        ; Odis
    dc.b %00100011,%10000110        ; Program
    dc.b %00100011,%10000000        ; Pr_ver
    dc.b %00100011,%10000011        ; Pr_inh
    dc.b %00000000,%00000010        ; Page_wr
    dc.b 0                          ; Manufacturer Code
    dc.b 0                          ; Device Code
    dc.b %00000000                  ; Tsmask
```

```
    dc.b 0                          ; Frei 3
    dc.b %00000000,%00000000        ; Frei 1
    dc.b %00000000,%00000000        ; Frei 2

    dc.b %00000000,%00001000        ; Tabelle A10
    dc.b %00000000,%00010000        ; A11
    dc.b %00000000,%00100000        ; A12
    dc.b %00000000,%01000000        ; A13
    dc.b %00000000,%00000000        ; A14
    dc.b %00000000,%00000000        ; A15
    dc.b %00000000,%00000000        ; A16

; Typ 27011, Index 11

    dc.b "27011 EPROM 8*16Kb*8 12.5V    "
                                    ; Beschreibung
    dc.l 131072   ; Kapazitaet in Bytes
    dc.l 16384    ; Groesse einer Seite
    dc.w 0        ; Programmieralgoritmus
    dc.w 10       ; i_zeit
    dc.w 30       ; m_zeit
    dc.w 25       ; trials
;         AAAAAAAA  BBBBBBBB
;         76543210  76543210

    dc.b %00010001,%10000011        ; Standby
```

```
    dc.b %00010011,%10000000        ; Read
    dc.b %00010011,%10000010        ; Odis
    dc.b %00110011,%00000010        ; Program
    dc.b %00110011,%10000000        ; Pr_ver
    dc.b %00110011,%10000011        ; Pr_inh
    dc.b %00000000,%00000010        ; Page_wr
    dc.b 0                          ; Manufacturer Code
    dc.b 0                          ; Device Code
    dc.b %00000000                  ; Tsmask
    dc.b 0                          ; Frei 3
    dc.b %00000000,%00000000        ; Frei 1
    dc.b %00000000,%00000000        ; Frei 2

    dc.b %00000000,%00001000        ; Tabelle A10
    dc.b %00000000,%00010000        ; A11
    dc.b %00000000,%00100000        ; A12
    dc.b %00000000,%01000000        ; A13
    dc.b %00000000,%00000000        ; A14
    dc.b %00000000,%00000000        ; A15
    dc.b %00000000,%00000000        ; A16
```

Listing 2: Die Typentabelle "typtab.asm" der Juniorsoft-Funktionen
für Aztec Assembler.

```
/* Programm zum Testen der Junior Prommer Hardware
 * (c) Merlin Computer   KICKSTART
 * erstellt am 18.4.88
 * Gerald Carda
 *
 * Aztec C 3.6
 * cc pruf.c +C +D
 * as junior -N -C -D
 * as typtab -N -C -D
 * ln pruf.o junior.o typtab.o -lcl
 */

#include "junior.h"

struct pruf
{
  char *IDString;
  UWORD SraDaten;
  UWORD SrbDaten;
};

struct pruf prufarray[24] =
{
  {(char *)"Abgl. der Prog.Sp. 25 V  Pin 1",144,0},
  {(char *)"Spannungstest      21 V  Pin 1",80,0},
  {(char *)"Spannungstest    12.5 V  Pin 1",48,0},
  {(char *)"Spannungstest       5 V  Pin 1",16,0},
  {(char *)"Spannungstest     4.3 V  Pin 1",1,0},
  {(char *)"Spannungstest       0 V  Pin 1",0,0},
  {(char *)"Spannungstest       5 V  Pin 2",0,32},
  {(char *)"Spannungstest       0 V  Pin 2",0,0},
  {(char *)"Spannungstest       5 V  Pin 20",0,1},
  {(char *)"Spannungstest       0 V  Pin 20",0,0},
  {(char *)"Spannungstest       5 V  Pin 21",0,8},
  {(char *)"Spannungstest       0 V  Pin 21",0,0},
  {(char *)"Spannungstest      25 V  Pin 22",128,4},
  {(char *)"Spannungstest     4.3 V  Pin 22",0,2},
  {(char *)"Spannungstest       0 V  Pin 22",0,0},
  {(char *)"Spannungstest      25 V  Pin 23",136,0},
  {(char *)"Spannungstest     4.3 V  Pin 23",0,16},
  {(char *)"Spannungstest       0 V  Pin 23",0,0},
  {(char *)"Spannungstest       5 V  Pin 26",0,0},
  {(char *)"Spannungstest       0 V  Pin 26",0,64},
  {(char *)"Spannungstest       5 V  Pin 27",0,128},
  {(char *)"Spannungstest       0 V  Pin 27",0,0},
  {(char *)"LED Test              LED leuchtet",2,0},
  {(char *)"LED Test              LED aus",0,0}
};

main()
{
  int i,error;

 if(GetParPort()!=0)
     {
      puts("Parallel Port belegt");
      exit(10);
     }
   if(GetTimer()!=0)
     {
      puts("Timer nicht erreichbar !?");
      exit(15);
     }

 puts("Funktiontest Junior Prommer Hardware");
 puts("");
 puts("Nach erfolgter Messung weiter mit <RETURN>");
 puts("Bitte KEIN Eprom einsetzen");
 puts("");
 puts("<RETURN> -> Start");
 getchar();

 for(i=0;i<=23;i++)
   {
    printf("Test Nr. %d %s   \n",i,
            prufarray[i].IDString);
    Junior((UWORD)0,
           (UWORD)prufarray[i].SraDaten,(UWORD)1);
    Junior((UWORD)0,
              (UWORD)prufarray[i].SrbDaten,(UWORD)2);
    puts("");
    puts("Weiter mit <RETURN>");
    getchar();
   }

 FreeTimer();
 FreeParPort();
 exit(0);
}
```

Listing 3:

Das C-Programm "pruf.c" zum Prüfen und Kalibrieren
des Juniorprommers.

```
/* Beispiel Programm (dupliziert ein 27512)
 * (c) Merlin Computer   KICKSTART
 * erstellt am 18.4.88
 * Gerald Carda
 *
 * Aztec C 3.6
 * cc beispiel +C +D
 * as junior -N -C -D
 * as typtab -N -C -D
 * ln beispiel.o junior.o typtab.o -lcl
 */

#include "junior.h"

main()
 {
   int i,error;
   ULONG juniorerror;
   UBYTE *speicher;

   EpromTyp=&TypTab; /* Basisadresse TypTab holen */
   EpromTyp+=9;      /* Eprom Typ auf 27512 (Index 9)
                      * setzen */

   if(GetParPort()!=0)
                     /* Parallel Port reservieren */
      {
       puts("Parallel Port belegt");
       exit(10);
      }
   if(GetTimer()!=0)    /* Timer oeffnen */
      {
       FreeParPort();
       puts("Timer nicht erreichbar !?");
       exit(15);
      }
   speicher=(UBYTE *)AllocMem(EpromTyp->et_Kapazi,
                              MEMF_PUBLIC);
   if(speicher==NULL)
      {
       FreeParPort();
       FreeTimer();
       puts("nicht genug Speicher");
       exit(20);
      }

   printf("EpromTyp  Name : %s Kapazität : %ld\n",
          EpromTyp->et_Zkette,EpromTyp->et_Kapazi);

 /* setze Typ */
   Junior((UWORD)10,(struct EpromTyp *)EpromTyp);

 /* init Prommer */
   Junior((UWORD)11);

   puts("Bitte Eprom des angezeigten Typs einsetzen
          <Return>");
   getchar();

 /* Lese Eprom nach Speicher von Eprom Adresse bis
    Eprom Adresse */
   juniorerror=Junior((UWORD)4,(ULONG)speicher,
                   (ULONG)0,(ULONG)EpromTyp->et_Kapazi);
   printf("Lesen    Ergebniss : %ld\n",juniorerror);

   puts("Bitte das Eprom fuer das Duplikat einsetzen
          <Return>");
   getchar();

   puts("Leertest");
   juniorerror=Junior((UWORD)5,
                   (ULONG)0,(ULONG)EpromTyp->et_Kapazi);
   printf("Leertest Ergebniss : %ld\n",juniorerror);

   puts("Brennen der Daten");
   juniorerror=Junior((UWORD)8,(ULONG)speicher,
                   (ULONG)0,(ULONG)EpromTyp->et_Kapazi,
                     (UWORD)0);
   printf("Brennen Ergebniss  : %ld\n",juniorerror);

   FreeParPort();
   FreeTimer();
   FreeMem(speicher,EpromTyp->et_Kapazi);
   exit(0);
 }
```

Listing 4: Das Beispielsprogramm "beispiel.c", welches den Gebrauch der Junior Prommer Funktionen zeigt.

```
/* Junior Prommer Header File
 * (c) Merlin Computer        KICKSTART
 * Gerald Carda
 * erstellt am 18.4.88
 */

#include "exec/types.h"
#include "exec/nodes.h"
#include "exec/types.h"
#include "exec/ports.h"
#include "exec/memory.h"
#include "exec/io.h"
#include "exec/libraries.h"
#include "exec/tasks.h"
#include "exec/execbase.h"
#include "exec/devices.h"
#include "devices/parallel.h"
#include "devices/timer.h"
#include <functions.h>

struct MsgPort *parport,*timeport;
struct IOExtPar *parreq;
struct timerequest *timereq;

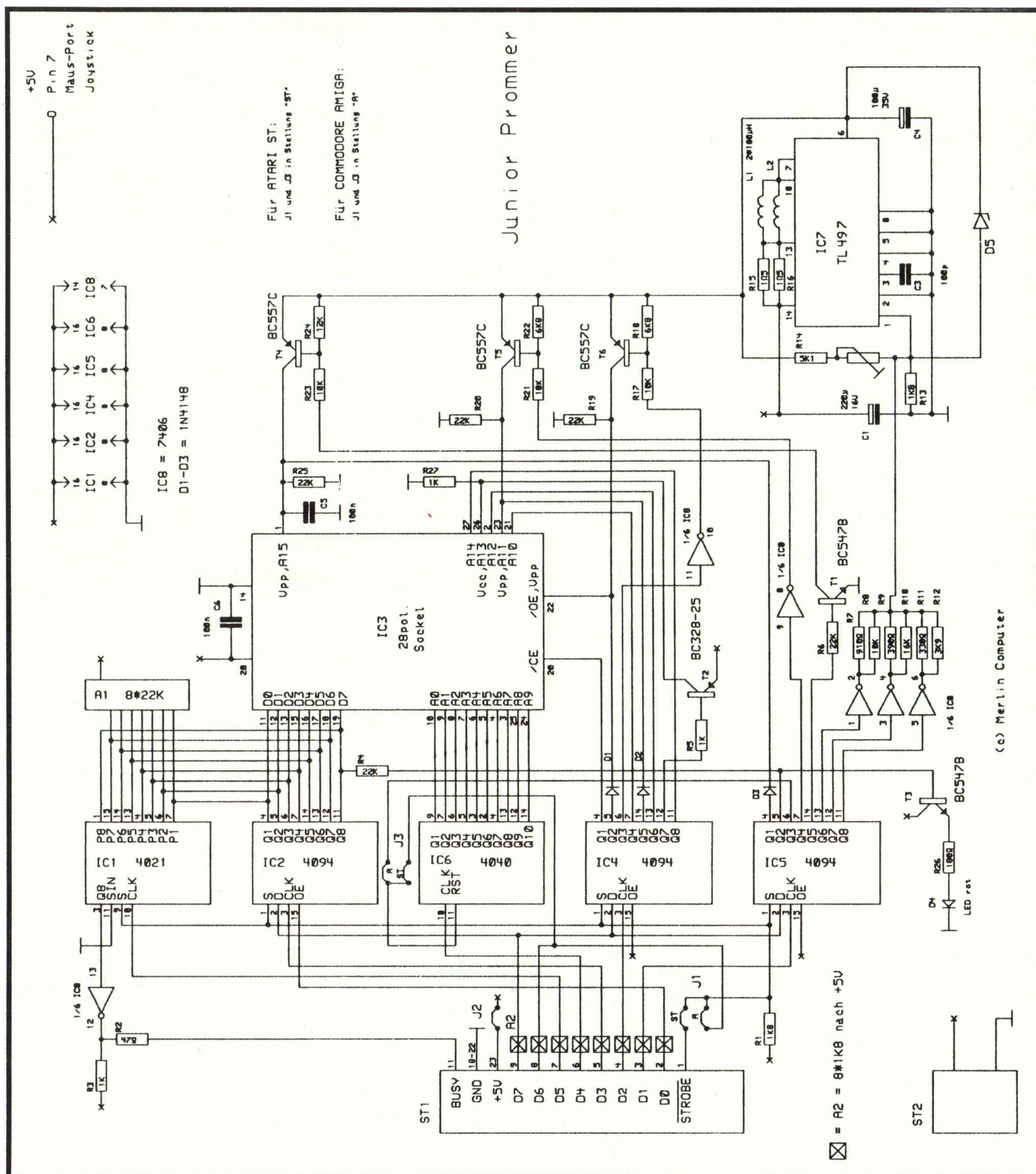
struct EpromTyp
 {
   char  et_Zkette[30];
   ULONG et_Kapazi;
   ULONG et_Seite;
   UWORD et_Alg;
   UWORD et_I_Zeit;
   UWORD et_M_Zeit;
   UWORD et_Trails;
   UWORD et_Standby;
   UWORD et_Read;
   UWORD et_Odis;
   UWORD et_Program;
   UWORD et_Pr_ver;
   UWORD et_Pr_inh;
   UWORD et_Page_wr;
   UBYTE et_Manufac;
   UBYTE et_Devcode;
   UBYTE et_Tsmask;
   UBYTE et_Frei3;
   UWORD et_Frei1;
   UWORD et_Frei2;
   UWORD et_M_a10;
   UWORD et_M_a11;
   UWORD et_M_a12;
   UWORD et_M_a13;
   UWORD et_M_a14;
   UWORD et_M_a15;
   UWORD et_M_a16;
 };

extern struct MsgPort *CreatePort();
extern struct IORequest *CreateExtIO();
extern struct IOStdReq *CreateStdIO();
struct EpromTyp *EpromTyp;
extern struct EpromTyp TypTab;

extern ULONG Junior();

GetParPort()
```

Der Schaltplan des JUNIOR PROMMER



Nachdem in der letzten Ausgabe der Abdruck des Schaltplans etwas dünn geraten ist, zeigen wir hier noch einmal den kompletten Schaltplan.

Als wichtigste Änderung ist hier die Diode D5 anzusprechen. D5 ist eine Z-Diode mit einer Zener-Spannung von 27 Volt. Mit Hilfe dieser Schaltungsänderung wird die maximale Ausgangsspannung des TL 497 auf 27 Volt begrenzt. Die Bauteile C4, T4, T5 und T6 sind dadurch vor Überspannung geschützt.

Die Jumper-Stellung für den Betrieb am AMIGA können nun ebenfalls aus dem Schaltbild abgelesen werden.

Wenn Sie die Platine nicht selbst herstellen wollen, können Sie eine fertig geätzte und gebohrte Platine bei der Redaktion bestellen. Die Platinen sind geprüft und mit Lötstopplack versehen. Im Lieferumfang der Platine ist außerdem die komplette Steuersoftware enthalten. Der Preis liegt bei DM 59.-. Der komplett aufgebaute Prommer ist für DM 229.- erhältlich.

```c
{
    int error;

    parport = CreatePort(0L,0L);
    if(parport == 0) return(-100);
    parreq =(struct IOExtPar *)
            CreateExtIO(parport,
                (long)sizeof(struct IOExtPar));
    if(parreq == 0)
        {
        DeletePort(parport);
        return(-200);
        }
    error = OpenDevice((UBYTE *)"parallel.device",
            0L,parreq,0L);
    if(error!=0)
        {
        DeleteExtIO(parreq,
            (long)sizeof(struct IOExtPar));
        DeletePort(parport);
        return(-300);
        }
    return(error);
}

FreeParPort()
{
    CloseDevice(parreq);
    DeleteExtIO(parreq, (long)sizeof(struct IOExtPar));
    DeletePort(parport);
}


GetTimer()
{
    int error;

    timeport = CreatePort(0L,0L);
    if(timeport == 0) return(-100);
    timereq =(struct timerequest *)
     AllocMem((long)
        sizeof(struct timerequest),MEMF_PUBLIC);
    if(timereq == 0)
        {
        DeletePort(timeport);
        return(-200);
        }
    timereq->tr_node.io_Message.
                        mn_Node.ln_Type=NT_MESSAGE;
    timereq->tr_node.io_Message.mn_Node.ln_Pri=0;
    timereq->tr_node.io_Message.mn_ReplyPort=timeport;
    error = OpenDevice(TIMERNAME,UNIT_MICROHZ,
                        timereq,0L);
    if(error!=0)
        {
        FreeMem(timereq,(long)
            sizeof(struct timerequest));
        DeletePort(timeport);
        return(-300);
        }
    return(error);
}

FreeTimer()
{
    CloseDevice(timereq);
    FreeMem(timereq,(long)sizeof(struct timerequest));
    DeletePort(timeport);
}

WaitForTimer(micro)
ULONG micro;
{
    timereq->tr_node.io_Command=TR_ADDREQUEST;
    timereq->tr_time.tv_secs=0L;
    timereq->tr_time.tv_micro=micro;
    DoIO(timereq);
}
```

Listing 5:
Das Headerfile "junior.h", welches alle Deklarationen für C-Programme enthält die den Junior Prommer benutzen wollen.

Schnell, komfortabel und erfolgreich programmieren mit Turbo-Pascal

Dieses neue Nachschlagewerk bringt Ihnen:

- **das Know-how zum Turbo-Pascal-System:** Turbo-Pascal-Editor, Programmierkurs (Schleifen, IF- und CASE-Anweisungen, Felder, Datenstrukturen, Prozeduren, Schnittstellentechnik, Dateiverarbeitung, Tips und Tricks (Reaktion auf Laufzeitfehler, Turbo-MSG automatisch laden...);
- **den Lehrgang Softwareengineering:** an einem Programmierprojekt zur Lagerverwaltung erfahren Sie den optimalen Weg von der Idee bis zum fertigen Programm;
- **Betriebssystemroutinen:** u.a. ROM-BIOS und IBM-Hardware (asynchrone Datenübertragung, Routinen zur Druckeransteuerung), MS-DOS-Aufrufe, Platz für mehr als 64 KB Dateien...
- **hardwarenahe Programmierung:** Rechnerkopplung, Floppytuning, Diskettendump, Druckersteuerung für den NEC P6...
- **raffinierte Utilities: Schnittstellenutilities** (Turbo Inliner, Anpassen von Word-Star-Dateien...), **Grafikutilities** (Zeichensatz-Editor, digitale Bildverarbeitung...), **Dateiverwaltungs-Utilities** (Ver-/Entschlüsseln von Dateien...);
- **den Turbo-Pascal-Assembler SIGMASM** zur automatischen Einbindung von Assembler-Routinen;
- **Grafik-Editor** zum Zeichnen und Ausdrucken von Linien oder Rechtecken; Invertieren, Spiegeln, Verschieben, Füllen von Flächen;
- **Anwenderprogramme für Technik/Mathematik:** u.a. lineare Algebra, statistische Qualitätskontrolle, **betriebswirtschaftliche Anwendungen:** z.B. Restschuldermittlung für Darlehen im Vergleich, **sonstige Anwendungen:** z.B. KI, Vokabeltraining;
- **Turbo-Pascal-Modulbibliotheken:** standardisierte Benutzeroberfläche mit Masken-/Menügenerator, Windowtechnik, Grafiksystem GSX inkl. Unterprogrammsammlung (z.B. Plotten zweidimensionaler Datensätze), Grafiktreiber HERC für die Herculeskarte;
- **Ergänzungsausgaben zum Grundwerk** mit neuen Utilities, Tools und Musterprogrammen.

Mit zusätzlichem Diskettenservice!

Musterprogramme, Utilities und Modulbibliotheken für raffinierte Lösungen in Turbo-Pascal und Weiterentwicklungen

Alle Programme auch unter Turbo-Pascal 4.0 lauffähig

Fordern Sie noch heute an:

Musterprogramme, Utilities und Modulbibliotheken für raffinierte Lösungen in Turbo-Pascal

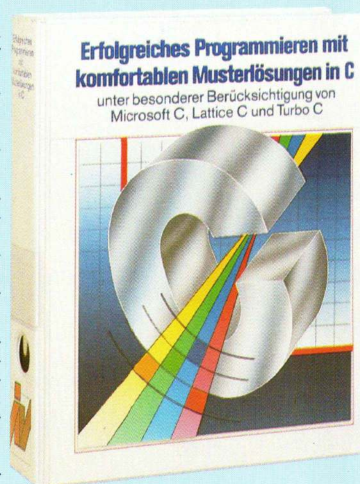
stabiler Kunstlederordner, DIN A4, ca. 800 Seiten, Bestell-Nr. 2500, Preis: DM 92,–.

Alle 2–3 Monate erhalten Sie Ergänzungsausgaben zum Grundwerk mit je ca. 120 Seiten zum Seitenpreis von 38 Pfennig (Abbestellung jederzeit möglich).

C-Musterlösungen für IBM PCs und Kompatible

Dieses Nachschlagewerk bietet Ihnen:

- **das Know-how zur Softwareerstellung in C:** Anhand von Syntaxdiagrammen und Beispielen lernen Sie das C-System detailliert kennen. Ein Schwerpunkt liegt auf den speziellen Compileroptionen von **Microsoft-** und **Turbo-C.** Bei Ihrer Programmentwicklung helfen Ihnen u.a. Kurse zu Grafik und Dateiverwaltung sowie der Praxisteil „Softwareengineering";
- **Utilities zur hardwarenahen Programmierung:** MS-DOS-Schnittstellenroutinen zum Einbinden von Betriebssystemaufrufen, RAM-Sortierroutinen, Zugriff des ROM-BIOS auf die serielle Schnittstelle, Interruptserviceroutine...;
- **komplette Anwender- und Funktionsbibliotheken zu Technik** (Meßwerterfassung, -verarbeitung, statistische Funktionen für Regressions-, Faktorenanalyse), **Wirtschaft/Verwaltung** (schnelles Indexverwaltungssystem), Full-screen-Editor TURBOSCRIPT sowie Bausteine für eine **Datei-Ein-/Ausgabe-** und eine **Grafikbibliothek;**
- **den „Desktop-Menü-Conductor"**, eine komfortable Benutzerschnittstelle mit Pull-down-Menüs und ausgefeilter Windowtechnik;
- **Add-on-Lösungen zu WordStar, dBase und Lotus 1-2-3,** mit denen Sie sich z.B. Ihre Programme für Artikel-, Lagerverwaltung und Gehaltsabrechnung maßschneidern;
- **einen zusätzlichen Disketten-service;**
- **Ergänzungsausgaben zum Grundwerk** mit Erweiterungen Ihrer Anwender- und Funktionsbibliotheken, Betriebssystem- und Assemblerroutinen sowie neuen Tools und Utilities.



Fordern Sie noch heute an:

Erfolgreiches Programmieren mit komfortablen Musterlösungen in C

stabiler Ringbuchordner, DIN A4, Grundwerk ca. 450 Seiten, Bestell-Nr. 3600, Preis: DM 92,–.

Alle 2–3 Monate erhalten Sie Ergänzungsausgaben zum Grundwerk mit je ca. 120 Seiten zum Seitenpreis von 38 Pfennig (Abbestellung jederzeit möglich).

Bitte mit 60 Pfennig freimachen

Postkarte/Antwort

INTEREST-VERLAG
Fachverlag für anspruchsvolle Freizeitgestaltung
z. Hd. Herrn Michel
Industriestraße 21
D-8901 Kissing

IN 160052

Meine Anschrift

Name, Vorname

Straße, Haus-Nr.

PLZ Wohnort

Nutzen Sie unser zusätzliches Angebot

___ Expl. **Erfolgreiches Hard- und Softwaretuning der Atari ST 260/520/1040 und Weiterentwicklungen**



mit

- ausführlichen Dokumentationen des 68000er-Prozessors und seiner Peripheriebausteine
- Aufbau, Funktionen und Applikationen des Atari ST
- Bauanleitungen (z.B. Hardwareerweiterungen)
- Programmierkurse (Pascal, C)
- Utilities, Tips und Tricks sowie
- Ergänzungsausgaben zum Grundwerk mit aktuellen Hardwaredokumentationen, Know-how zum Atari ST, Programmiersprachkursen und neuen Utilities.

Stabiler Ringbuchordner, DIN A 4, ca. 450 Seiten, Bestell-Nr. 3700, Preis: DM 92,–.
Alle 2–3 Monate erhalte ich bis auf Widerruf die Ergänzungsausgaben zum Grundwerk mit jeweils ca. 120 Seiten zum Seitenpreis von 38 Pfennig.

Konzentriertes IC-Wissen für Hardwareerweiterungen und -entwicklungen

Halbleiterbausteine beherrschen heute die Welt der Elektronik. Der IC-Markt ist selbst für Fachleute nicht mehr überschaubar, und laufend nimmt das Angebot an integrierten Schaltungen weiter zu!

Hier setzt unser **„Aktuelles IC-Datenbuch"** an. Ob Reparatur oder Neuentwicklung – dem Hobbyelektroniker wie dem Praktiker dient es als unentbehrlicher Ratgeber.

- **Zu den digitalen und analogen ICs** gibt Ihnen das Werk u.a. folgende Daten: Pinbelegung, Leistungsaufnahme. Eingangsimpedanz, Ausgangsbelastbarkeit, Typvarianten, internes- oder Blockschaltbild, Temperaturbereiche, Schaltpegel, Kaltkapazitäten, Applikationsbeispiele, Herstellerfirmen:
- **Zusätzlich bei Computerbausteinen:** Schaltungsapplikation und Testschaltung, Beschreibung der einzelnen Funktionen, bei Mikroprozessoren der vollständige Befehlssatz mit Erläuterungen, Taktfrequenzen, Logiksymbole, Signal-Zeit-Diagramme, Verweise zu äquivalenten Typen, Bezugsquellen, bei Speichern Angaben zum Programmierverfahren und Zugriffszeiten.

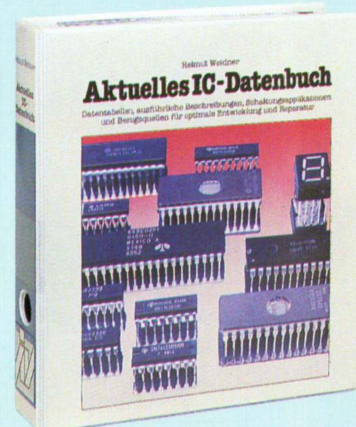
Aus dem Inhalt:
digitale und halblineare ICs: TTL- und CMOS-Logik-ICs, Prozessoren, Speicherbausteine, A/D-Wandler...:
lineare ICs: Spannungsregler, Operationsverstärker, NF-/HF-Verstärker, Radio-/TV-Schaltkreise, Fernsteuer-ICs, Sensoren, Funktionsgeneratoren u.a.

Fordern Sie noch heute an:

Aktuelles IC-Datenbuch

stabiler Ringbuchordner, DIN A4, über 1000 Seiten, Bestell-Nr. 1500, Preis: DM 92,–.

Alle 2–3 Monate wird dieses Werk mit den neuesten Daten inkl. ausführlichen Beschreibungen aktualisiert (jeweils ca. 120 Seiten zum Seitenpreis von 38 Pfennig) (Abbestellung jederzeit möglich).





Industriestraße 21
D-8901 Kissing
Tel. 08233/2101-0

INTEREST-VERLAG
Fachverlag für anspruchsvolle Freizeitgestaltung

VON HERBERT KUNZ

MIT BASIC ANS SYSTEM



Teil 4: Rund um die Maus

Was wäre der Amiga ohne die grafische Benutzeroberfläche Intuition! Das wichtigste Instrument zur Nutzung von Intuition ist die Maus. Wenn Sie diese Maus über den Schreibtisch bewegen, so folgt auf dem Bildschirm ein Pfeil jeder Bewegung. Mit diesem Mauszeiger (unter Intuition "Pointer" genannt) wollen wir uns etwas näher beschäftigen.

Normalerweise hat der Mauszeiger nach dem Booten der Workbench immer dasselbe Aussehen. Vielleicht haben Sie ihm schon einmal mit dem Programm Preferences ein anderes Aussehen verliehen. Von Amiga Basic aus aber ist es nicht ohne weiteres möglich, diesen Zeiger zu ändern. Sicher haben Sie auch schon einmal ein Anwenderprogramm gesehen, in dem der Mauszeiger während der Programmausführung sein Ausehen verändert; denken Sie nur an ein Zeichenprogramm, das Ihnen verschiedene Pinselformen zur Verfügung stellt. Auch Sie können diese Möglichkeit nutzen, wenn Sie die Library-Routinen benutzen. Zum Ändern des Mauszeigers gibt es mehrere Möglichkeiten. Eine davon ist der Aufruf der Routine SetPointer(). Diese Routine hat folgendes Format:

```
CALL Set Pointer (window,
pointer, height, widht, xOffset,
yOffset)
```

window = Ein Zeiger auf die Windowstruktur des Fensters, in dem der neue Zeiger zu sehen sein soll.

pointer = Ein Zeiger auf die Daten des Mauszeigers.

height = Die Höhe des Mauszeigers.

widht = Die Breite des Mauszeigers.

xOffset und yOffset = Die Werte des"HotSpots".

Diese Werte bestimmen den Punkt, an dem der Mausklick wirkt, und müssen von der Zahl 255 abgezogen werden. Der Wert 255 ist die linke obere Ecke. Diese Routine stellt den neuen Zeiger nur in dem angegeben Fenster dar.

Nur in einem Fenster!

Wird ein anderes Fenster angeklickt, so erscheint wieder der Standardzeiger. Um im ausgewählten Fenster wieder den Standardzeiger darzustellen, gibt es folgende Routine:

```
CALL ClearPointer (window)
```

In "window" geben Sie wieder den Zeiger auf die Windowstruktur an.

Beim Mauszeiger handelt es sich übrigens um ein Sprite. Sie wissen ja, ein Sprite ist ein grafisches Objekt, das unabhängig von der übrigen Bildschirmanzeige bewegt werden kann. Diese Bewegung des Mauszeigers wollen wir jetzt einmal programmgesteuert vornehmen. Dazu müssen wir noch einmal die Graphics Library bemühen. Zuerst müssen wir uns eine Simple Sprite-Struktur anlegen. Diese Struktur hat folgendes Aussehen:

```
4 Bytes - Zeiger auf Spritedaten
2 Bytes - Höhe des Sprites
2 Bytes - X-Position des Sprites
2 Bytes - Y-Position des Sprites
2 Bytes - Nummer des Sprites
```

Nachdem diese Struktur angelegt wurde, können wir mit folgender Routine ein Sprite bewegen:

```
CALL MoveSprite (viewport,
simplesprite,x,y)

viewport = Zeiger auf die
       Viewportstruktur

simplesprite = Zeiger auf die
       Simple Sprite-Struktur

x,y = Die x- und y-Koordina-
       ten, an denen das
       Sprite dargestellt
       werden soll.
```

Anhand von Listing 1 können Sie sich die Programmierung eines eigenen Mauszeigers veranschaulichen. Die Daten des neuen Zeigers werden in Datazeilen gespeichert. Die ersten und letzten beiden Werte der Spritedaten müssen den Wert 0 haben, jeder Punkt kann eine von 4 Farben annehmen. Wenn Sie das Zeichen "-" einsetzen, so wird die Hintergrundfarbe genommen. Das Sprite ist also an dieser Stelle durchsichtig. Eine "1" steht für Farbe 1, eine "2" für Farbe 2, und für Farbe 3 setzen Sie eine "3" ein. Die Farben für dieses Sprite werden aus den Farbregistern 17-19 genommen; diese Farben können Sie mit dem Befehl PALETTE verändern. Wird das Sprite nicht mehr benötigt, so teilen Sie das dem System mit der Routine FreeSprite mit.

```
CALL FreeSprite (spritenum)

spritenum = die Nummer des Sprites.
```

Mausvielfalt

Nun zu der zweiten Möglichkeit, den Mauszeiger zu verändern. Mit Set-Pointer bestand ja nur die Möglichkeit, den Zeiger für ein Fenster zu verändern. Sie können aber den Zeiger auch dauerhaft verändern, ohne das Programm Preference zu benutzen, denn die Intuition Library stellt uns Routinen zur Änderung der Preference-Struktur zur Verfügung. In dieser Struktur sind die Werte zu finden, die Sie mit Preference eingestellt haben, also z.B. die Workbenchfarben, den angeschlossenen Drucker wie natürlich auch die Daten des Mauszeigers.

Struktur Menu

Offset	Byte	Bedeutung
0	L	Zeiger auf nächste Menustruktur
4	W	Linke Ecke der Menubox in Bildschirmpixel
6	W	Obere Ecke der Menubox in Bildschirmpixel
8	W	Breite der Menubox
10	W	Höhe der Menubox
12	W	Flags - 1 = Menü ist eingeschaltet 0 = Menü ist ausgeschaltet &H100 = setzt Intuition wenn Menükasten zu sehen ist.
14	L	Zeiger auf Text des Menükopfes
16	L	Zeiger auf erste Menuitem Struktur
20	4 * W	wird von Intuition benutzt

Struktur Menuitem

Offset	Byte	Bedeutung
0	L	Zeiger auf nächste Item-Struktur.
4	W	Linke Ecke des Items.
6	W	Obere Ecke des Items.
8	W	Breite des Items.
10	W	Höhe des Items.
12	W	Flags (siehe Tab. Flags)
14	L	Mutual Exclude - bestimmt die Items, die gleichzeitig eingeschaltet sein dürfen. Bit 0=Item 1, Bit 1=Item 2 u.s.w.
18	L	Zeiger auf Image- oder Intuitextstruktur. (wird nach Anwahl des Menüpunktes dargestellt)
22	L	Zeiger auf zweite Image- oder Intuitextstruktur (wird dargestellt, wenn der Mauszeiger über diesem Item steht)
26	B	Zeichen für Commandosequenz. Wird mit rechter Amigataste + Zeichen angewählt.
27	B	Damit nächster Wert an gerader Adresse.
28	L	Zeiger auf Subitem-Struktur. Wie Item-Struktur. Ein weiteres Subitem wird ignoriert.
32	W	wird von Intuition benutzt.

Intuitext - Struktur

1 Byte	Vordergrundfarbe
1 Byte	Hintergrundfarbe (für 0 Bits bei Zeichenmode JAM 2)
1 Byte	Zeichenmodus
1 Byte	Füllbyte
2 Byte	X-Koordinate des Textbeginns in Bildschirmpixel
2 Byte	Y-Koordinate des Textbeginns in Bildschirmpixel
4 Byte	Zeiger auf TextAttr Struktur (0=Standardfont)
4 Byte	Zeiger auf Text (Text mu_ mit einem Nullbyte enden)
4 Byte	Zeiger auf weitere Intuitext-Struktur

Flagbits für Menuitem

Wert	Bedeutung
1	Das Item wird nach dem Anwählen markiert. (mit dem Haken)
2	Das Item ist gleich zu Beginn markiert.
4	Ist zu setzen, wenn dieser Menüpunkt durch eine Tastenkombination angewählt werden kann.
8	Ein markiertes Item bleibt markiert.
16	Ist zu setzen, wenn das Item eingeschaltet ist. Ist das Item nicht eingeschaltet, wird es gepunktet dargestellt.

Zum Auslesen dieser Daten gibt es zwei Routinen:

```
CALL GetDefPrefs
(prefbuffer,size)
```

prefbuffer = Zeiger auf Speicherbereich, in dem die gelesenen Daten geschrieben werden sollen.
size = Anzahl der zu lesenden Bytes.

Mit dieser Routine werden die Daten von der Diskette gelesen. Es gibt auch die Möglichkeit, die momentan eingestellten Werte aus dem Speicher zu lesen.

```
CALLGetPrefs (prefbuffer,size)
```

Die Parameter sind die gleichen wie bei GetDefPrefs.

Nachdem die Preference-Struktur eingelesen wurde, können Sie dort Änderungen vornehmen und diese Struktur wieder speichern. Dazu dient folgende Routine:

```
CALL SetPrefs (prefbuffer,
size,disk)
```

Die ersten beiden Parameter sind die gleichen wie vorher beschrieben. Für "disk" setzen Sie den Wert 1, wenn Sie die Daten auf Diskette schreiben wollen, oder den Wert 0, wenn die Daten ins Ram gespeichert werden sollen.

Listing 2 zeigt diese Möglichkeit zum Ändern des Mauszeigers.
Ein Fenster läßt sich mit dem Symbol in der rechten unteren Ecke des Fensters vergrößern oder verkleinern. Dieser Vergrößerung oder Verkleinerung sind aber Grenzen gesetzt. Zum Einstellen dieser Grenzen dient folgende Routine:

```
CALL WindowLimits (window,
min_widht,min_hight,max_widht,max_hight)
```

window = Ein Zeiger auf die Window-Struktur.
min_width = Die minimale Breite des Fensters.
min_hight = Die minimale Höhe des Fensters.
max_widht = Die maximale Breite des Fensters.
max_hight = Die maximale Breite des Fensters.

Die nächsten Bits beschreiben das Aussehen, wenn der Mauszeiger über dem Menuitem steht.

0	wenn kein Bit gesetzt ist, so wird ein alternativer Text oder ein zweites Image dargestellt.
64	bei diesem Wert wird das Item invertiert.
128	Das Item wird mit einem Rahmen umgeben.
192	Das Item verändert sich nicht.

Die nächsten Werte werden von Intuition gesetzt.

256	Das Item wurde angewählt
4096	Das Item ist momentan zu sehen.
8192	Der Mauszeiger steht über diesem Item.

```
 ' Benutzung eigener Mauszeiger
  ' in Amiga-Basic-Programmen
 '        H. Kunz  1988
 ' (c) KICKSTART 1988

DECLARE FUNCTION AllocMem& LIBRARY
DECLARE FUNCTION GetSprite& LIBRARY
LIBRARY "intuition.library"
LIBRARY "exec.library"
LIBRARY "graphics.library"

ActivateWindow WINDOW(7)          ' Ausgabefenster aktivieren

block&=2^1+2^16
speicher&=AllocMem& (112,block&) ' 112 Bytes reservieren
IF speicher&= 0 THEN ERROR 7
speicher1&=speicher&

GOSUB mausanlegen
SetPointer WINDOW(7),speicher1&,hoehe%,breite%,248,248

simpspr&=speicher1&+100
POKEL simpspr&,speicher1&         ' Zeiger auf Spritedaten
POKEW simpspr&+4,hoehe%           ' Hoehe des Sprites
POKEW simpspr&+6,0                ' X- Koordinate
POKEW simpspr&+8,0                ' Y- Koordinate
POKEW simpspr&+10,1               ' Sprite Nummer
sn&=1:sn&=GetSprite& (simpspr&,sn&)

viewport&=PEEKL(WINDOW(7)+46)+44:y&=120
 FOR x&=1 TO 550                          ' Mauszeiger bewegen
  MoveSprite viewport&,simpspr&,x&,y&
 NEXT
 FOR x&=550 TO 1 STEP -1
  MoveSprite viewport&,simpspr&,x&,y&
 NEXT

 FreeSprite sn&
 ClearPointer WINDOW(7)
 FreeMem speicher1&,112          ' Speicher zurueckgeben

 LIBRARY CLOSE                   ' Library schliessen
 END

mausanlegen:

  READ breite%,hoehe%,tiefe%
  FOR x=1 TO hoehe%
   READ a$
   plane1=0:plane2=0
    FOR i=1 TO breite%
     zeichen$=MID$(a$,i,1)

     IF zeichen$="1" OR zeichen$="3" THEN
       plane1=plane1+2^(16-i)
      END IF

       IF zeichen$="2" OR zeichen$="3" THEN
```

Jedes Fenster hat einen eigenen Namen. Dieser Name wird bei Aufruf des Basic-Befehls WINDOW übergeben. Mit folgender Routine läßt sich der Fenstername nachträglich ändern:

```
CALL SetWindowTitles (window,
windowtitle,screentitle)
```

window = Der Zeiger auf die Window-Struktur.

windowtitle = Ein Zeiger auf den Fenstertext.

screentitle = Ein Zeiger auf den Screentext.

Der Text muß mit einem Nullbyte abgeschlossen sein. Den Beginn des Textes ermitteln Sie mit dem Amiga Basic-Befehl SADD. Der Screentitel ändert sich jedesmal, wenn das angegebene Fenster aktiv ist. Sie können ein Fenster auch aktivieren, ohne es mit der Maus anzuklicken.

```
CALL ActivateWindow (window)
```

window = Zeiger auf die Window-Struktur.

Der Guru kommt!

Wenn Sie mit den Systemroutinen experimentiert haben, so ist Ihnen sicherlich schon einmal die Alarmmeldung des Amiga untergekommen, die GURU MEDITATION. Diese sogenannten Alerts können Sie auch programmieren. Folgende Routine dient dazu:

```
taste = DisplayAlert&
(alertnummer,string,height)
```

alertnummer = 0 = Recovery Alert,
&H8000 = Dead End Alert.

string = Zeiger auf den auszugebenden Text.

height = Die Höhe des Alert in Bildschirmpixel.

Ein Recovery Alert ist die harmlosere Form. Bei diesem Alert-Typ bekommt nach Drücken einer Maustaste Ihr Programm wieder die Programmkontrolle. Gleichzeitig können Sie noch er-

```
            plane2=plane2+2^(16-i)
          END IF

        NEXT
        POKEW speicher&,plane1:POKEW speicher&+2,plane2
        speicher&=speicher&+4
      NEXT
        POKE speicher1&+100,248
        POKE speicher1&+101,248

      DATA 16 :' Breite    muss 16 sein
      DATA 16 :' Hoehe
      DATA  2 :' Tiefe     muss 2 sein
      DATA ----------------
      DATA 11--11----22---2
      DATA 11--11----22--22
      DATA 11--11----22--22
      DATA 11--11----22--2-
      DATA 11--11----22-22-
      DATA 11--11----22-22-
      DATA 111111----222---
      DATA 111111----222---
      DATA 11--11----22-22-
      DATA 11--11----22-22-
      DATA 11--11----22--2-
      DATA 11--11----22--22
      DATA 11--11----22--22
      DATA 11--11----22---2
      DATA ----------------

    RETURN
```

Listing 1: Benutzung eigener Mauszeiger in AMIGA Basic-Programmen

```
' Benutzung eigener Mauszeiger
'  in Amiga-Basic-Programmen
'  durch Auslesen und Aendern
'  der Preference-Struktur
'       H. Kunz  1988
' (c) KICKSTART 1988

DECLARE FUNCTION AllocMem& LIBRARY
LIBRARY "intuition.library"
LIBRARY "exec.library"

block&=2^1+2^16
speicher&=AllocMem& (108,block&)
IF speicher&= 0 THEN ERROR 7
speicher1&=speicher&

 ' speicher&+28  = Beginn der Daten des Mauszeigers
 ' speicher&+100 = x-Offset des 'Hot Spots'
 ' speicher&+101 = y-Offset des 'Hot Spots'
 ' speicher&+102 = Farbregister 17
 ' speicher&+104 = Farbregister 18
 ' speicher&+106 = Farbregister 19

GetPrefs speicher1&,108      ' Daten lesen
GOSUB mausanlegen
SetPrefs speicher1&,108,0    ' Daten im Ram aendern

 FreeMem speicher1&,108      ' Speicher zurueckgeben

  LIBRARY CLOSE              ' Library schliessen
 END

mausanlegen:

   speicher&=speicher&+28
   READ breite%,hoehe%
   FOR x=1 TO hoehe%
     READ a$
```

mitteln, welche Maustaste gedrückt wurde. Steht in Taste der Wert 1, so wurde die linke, anderfalls die rechte Taste gedrückt. Vor dem Text steht als 16-Bit-Wert die X-Koordinate und als 8-Bit-Wert die Y-Koordinate, welche die Position des Texts bestimmen. Nach dem Text setzen Sie einen 16-Bit-Wert, der ungleich Null sein muß, wenn weiterer Text folgen soll. Ein Nullbyte schließt den Text ab. In Listing 3 können Sie die Programmierung eines Recovery - Alert sehen.

Menü 2 bitte!

Kommen wir nun zu den Menüs. Amiga Basic stellt auch hierfür Befehle zur Verfügung. Allerdings sind mit diesen Befehlen nicht alle Möglichkeiten der Menüprogrammierung abgedeckt. Wenn Sie Ihre Menüs mit allen Möglichkeiten ausstatten wollen, so müssen Sie sich intensiver mit der Menüstruktur und mit den Systemroutinen zur Menüprogrammierung beschäftigen. Die Menüstruktur können Sie aus Tab. 1 ersehen. Die Basisadresse der Menüstruktur finden Sie in der Windowstruktur; sie läßt sich folgendermaßen ermitteln:

```
Menustruktur = PEEKL (window
(7)+28)
```

Dazu noch ein paar Begriffserklärungen. Wenn Sie die rechte Maustaste drücken, so erscheinen in der Menüzeile die Menütexte. Wenn Sie mit gedrückter rechter Maustaste auf einen Menütext fahren, so rollen die Menueitems herunter. Manche Menueitems haben noch sogenannte Subitems. Diese sind zu sehen, wenn Sie mit dem Mauszeiger über einem Menueitem sind. Hier nun die Systemroutinen zur Menüprogrammierung:

```
CALL ClearMenuStrip (window)
```

```
window = Zeiger auf die Win-
dow-Struktur.
```

Damit werden sämtliche Menüs gelöscht, die zu diesem Fenster gehören.

```
menuitem = ItemAdress&
(menunr,itemnr)
```

```
menunr = Die Nummer des Menüs.
itemnr = Die Nummer des Items.
```

```
      plane1=0:plane2=0
      FOR i=1 TO breite%
        zeichen$=MID$(a$,i,1)

        IF zeichen$="1" OR zeichen$="3" THEN
          plane1=plane1+2^(16-i)
        END IF

        IF zeichen$="2" OR zeichen$="3" THEN
          plane2=plane2+2^(16-i)
        END IF

      NEXT
      POKEW speicher&,plane1:POKEW speicher&+2,plane2
      speicher&=speicher&+4
    NEXT
      POKE speicher1&+100,254
      POKE speicher1&+101,255

    DATA 16 :' Breite    muss 16 sein
    DATA 18 :' Hoehe     muss 18 sein.

    DATA ----------------
    DATA 111-------------
    DATA 333-------------
    DATA 333-------------
    DATA 333-------------
    DATA 333-------------
    DATA 333-------------
    DATA 333-------------
    DATA 333223322332233-
    DATA 3332233223322333
    DATA 3332233223322333
    DATA 3332233223322333
    DATA 3332233223322333
    DATA 3333333333333333
    DATA 3333333333333333
    DATA -33333333333333-
    DATA -333333333333---
    DATA ----------------

  RETURN
```

Listing 2: Benutzung eigener Mauszeiger durch Auslesen und Aendern der Preference-Struktur

```
 ' Alerts in Amiga-Basic-Programmen
 '    H. Kunz 1988
 ' (c) KICKSTART 1988

DECLARE FUNCTION DisplayAlert& LIBRARY
LIBRARY "intuition.library"

t1$=CHR$(0)+CHR$(50)+CHR$(15)  ' x+y Position 1. String
t2$=CHR$(0)+CHR$(50)+CHR$(28)  ' x+y Position 2. String
text$=t1$+"Keine Angst! Dies ist ein Recovery-Alert"
          +CHR$(0)+CHR$(1)
text$=text$+t2$+"Druecken Sie eine Maustaste"+CHR$(0)

num&=0  ' Recovery Alert - &H8000 fuer Dead-End-Alert
taste&=DisplayAlert& (num&,SADD(text$),40)

PRINT "Prima, Sie haben die ";
IF taste&=1 THEN
  PRINT "linke";
  ELSE
  PRINT "rechte";
END IF
PRINT " Taste gedrueckt !"

LIBRARY CLOSE
```

Listing 3: Alerts in AMIGA Basic-Programmen

Nach Rückkehr aus dieser Routine steht in "menuitem" ein Zeiger auf die Struktur des gesuchten Menueitems.

`CALL OffMenu (window,menunr)`
window = Zeiger auf die Window-Struktur. menunr = Nummer des Menüpunktes.

Mit dieser Routine wird ein Menüpunkt abgeschaltet.

`CALL OnMenu (window,nr)`
`Die Parameter sind dieselben wie bei OffMenu.`

Mit dieser Routine wird der Menüpunkt wieder eingeschaltet.

`CALLSetMenuStrip (window,menu)`
window = Zeiger auf die Window-Struktur. menu = Zeiger auf die Menüstruktur.

Damit wird ein Menü eingeschaltet.
Nun haben Sie die Informationen, die Sie zur Menüprogrammierung brauchen. Erstellen Sie sich doch einmal Menueitems, die nach dem Anwählen einen anderen Text darstellen, oder Subitems, die aus einer Zeichnung bestehen. Die Image-Struktur, die Sie dafür benötigen, wird später noch beschrieben.
Intuition kann Ihrem Programm auch Signale senden.

Informationen für Jedermann

Wenn Sie wissen wollen, ob ein Menüpunkt ausgewählt, das Fenster vergrößert oder eine Diskette gewechselt wurde, so können Sie den IDCMP abfragen (IDCMP = Intuition Direct Communication Message Port). In den IDCMP-Flags wird angegeben, bei welchen Funktionen Ihnen Intuition ein Signal senden soll. Diese Flags können Sie mit der Routine ModifyIDCMP ändern.

`CALL ModifyIDCMP (window,flags)`
window = Zeiger auf die Window-Struktur. flags = Die zu ändernden Flags.

In Listing 4 können Sie sehen, wie Sie den IDCMP abfragen und welche Flags möglich sind.

```
' Abfrage des IDCMP
' H. Kunz 1988
' (c) KICKSTART 1988

DECLARE FUNCTION GetMsg& LIBRARY
LIBRARY "exec.library"
LIBRARY "intuition.library"

flags&=PEEKL (WINDOW(7)+82)        ' Momentan gesetzte Flags
flags&=flags& OR 65536&           ' Diskremoved dazugeben
ModifyIDCMP WINDOW(7),flags&       ' Flags aendern

warte:
  userport&=PEEKL(WINDOW(7)+86)' Zeiger auf Userport
  signal& =GetMsg&(userport&) ' Message-Port ermitteln
  idcmp&  =PEEKL(signal&+20)   ' IDCMP-Flags lesen
  PRINT idcmp&:
  IF idcmp&=65536& THEN aus     ' Keine Disk im Laufwerk
                                ' Programm beenden
GOTO warte

aus:
  LIBRARY CLOSE

' IDCMP Flags

'  Wert      Bedeutung

'       1     Sizeverify
'       2    Fenstergroesse veraendert
'       4     Refreshwindow
'       8    linke Maustaste wurde gedrueckt
'      16     Mausbewegung
'      32     Gadget down
'      64     Gadget up
'     128     Reqset
'     256     Menu angewaehlt
'     512    Fenster wurde geschlossen
'    1024     Taste gedrueckt
'    2048     Reqverify
'    4096     Reqclear
'    8192     Menuverify
'   16384    Neue Preference-Struktur erstellt
'   32768     Diskette eingelegt
'   65536     Diskette entnommen
'  262144     Fenster aktiv
'  524288     Fenster nicht aktiv
'
```

Listing 4: Abfrage des IDCMP

Durch Addieren der Flagwerte erhalten Sie mehrere Möglichkeiten der Kombination einzelner Flags.

Zum Zeichenstift

In der Intuition Library gibt es eine Routine, mit der zusammenhängende Linienpaare gezeichnet werden können.

`CALL DrawBorder (rastport, border,x-offset,y-offset)`
`rastport = Zeiger auf die Rastportstruktur.` **border = Zeiger auf eine Border-Struktur.** **x-offset = Beginn der X-Position von der linken oberen Ecke aus.** **y-offset = Beginn der Y-Position von der linken oberen Ecke aus.**

```
Die Borderstruktur hat folgendes Aussehen:

2 Bytes = horizontaler Abstand von x- und y-offset
2 Bytes = vertikaler Abstand von x- und y-offset
1 Byte  = Zeichenfarbe
1 Byte  = Hintergrundfarbe (nicht benutzt)
1 Byte  = Zeichenmodus
1 Byte  = Anzahl der x-y Paare
4 Bytes = Zeiger auf die Koordinaten
4 Bytes = Zeiger auf eine weitere Borderstruktur
          (0, wenn keine weitere Struktur existiert)
```

```
Die Image-Struktur hat folgendes Aussehen:

2 Bytes  = X-Abstand zu x- und y-offset.
2 Bytes  = Y-Abstand zu x- und y-offset.
2 Bytes  = Breite des Image.
2 Bytes  = Höhe des Image.
4 Bytes  = Zeiger auf Imagedaten.
1 Byte   = Bitplanes, in denen  gezeichnet wird.
1 Byte   = Hintergrundfarbe.
4 Bytes  = Zeiger auf evtl. weitere Image-Struktur.
```

Die Koordinatenwerte sind keine absoluten Werte, sondern gelten relativ zur Anfangsposition.
Zum Darstellen von Zeichnungen gibt es unter Intuition folgende Routine:

```
CALL DramImage(rastport,image,
x-offset,y-offset)
```

rastport = Zeiger auf die Rastport-Struktur.

image = Zeiger auf die Image-Struktur.

x-offset = Beginn der X-Position von der linken oberen Ecke aus.

y-offset = Beginn der Y-Position von der linken oberen Ecke aus.

Ein Beispiel für ein Image sind die Icons der Workbench.
Wenn Sie sich ein Image für die Menüstruktur erstellen wollen, so können Sie auch die Image-Daten in Datazeilen packen und die Routine "mausanlegen" aus Listing 1 und 2 benutzen.
Viel Information, liebe Leser, probieren Sie das eine oder andere ruhig aus. Experimentiern Sie auch ein wenig herum, Sie werden schnell mehr Einblick in die Prizipien von Intuition erhalten.

Damit sind wir für dieses mal wieder am Schluß angelangt. Das nächste mal werden wir uns ausführlich mit den Gadgets und den Routinen für die Workbench beschäftigen.

ENDE

VON LARS SCHMITT

MODEM TALKING

Teil 1: DFÜ, was ist das?

Es ist noch nicht so lange her, als - kurz nach der alljährlichen weihnachtlichen Bescherung - der Vater mit dem Sohne in dessen Zimmer verschwand, um mit der neuen Eisenbahn zu spielen. Doch die guten alten Zeiten sind endgültig vorbei. Noch vor kurzer Zeit wunderten sich die Eltern, wie sich der Sohn oder die Tochter Stund' um Stund' mit einem kleinen grauen Kästchen mit vielen nichtssagenden Tasten beschäftigen kann, warum die Sesamstraße auf einmal egal ist, und selbst die frisch gekochten Miracoli nebensächlich sind. Neuerdings geraten die Regierungen unserer lieben Kleinen vollends aus dem Häuschen, wenn aus den heimischen Kinderstuben Gepiepse und Gepfeife dringt. Spätestens, wenn der Vater auf die Frage: "Haste ein Vogel?" die Antwort "Nein, das ist mein neues Modem, 1200 Baud Vollduplex, mit FTZ-Nummer und RS 232C-Schnittstelle!" bekommt, ist es passiert: Ratlosigkeit. Da hilft nur eines: WEITERLESEN!

In dieser und in den folgenden Ausgaben steht an dieser Stelle jeweils ein Bericht über DFÜ (übrigens, das bedeutet: DATEN FERN ÜBERTRAGUNG). Heute erkläre ich Ihnen die Grundlagen, also wie die DFÜ überhaupt funktioniert.

Computersignale laufen meistens über Kabel. Welche Kabel bieten sich da besser an, als die Telefonkabel? Die sind nämlich überall gegenwärtig, und fast jeder Mensch hat auch ein Telefon. Deshalb läuft heute ein Großteil der DFÜ über das ganz normale Telefonnetz ab. Aber - wie immer, wenn es gerade schön wird - fangen hier die Probleme schon an:

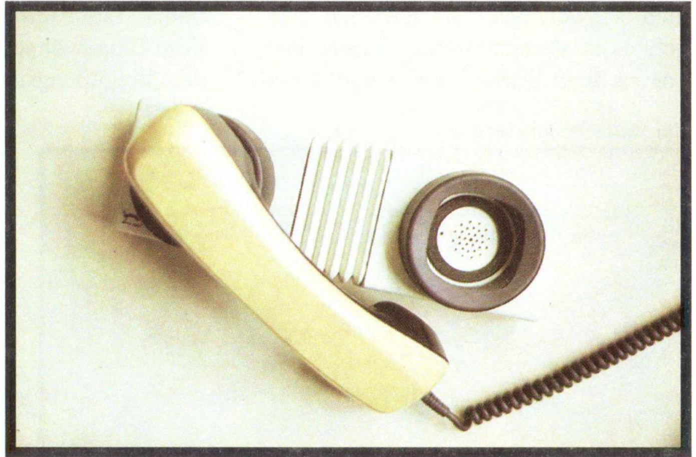
Problem 1:

Wie bekomme ich aus den Stromimpulsen eines Computers hörbare Töne, denn nur die kann man ja über die Telefonleitungen senden?

Lösung:

Man benutzt ein MODEM. MODEM ist ein Kunstwort, zusammengesetzt aus MODULATOR/DEMODULATOR. So ein MODEM kann also die elektrischen Impulse des Computers in hörbare Töne umwandeln. Man kann sich ein MODEM am besten als kleinen Kasten vorstellen. Auf der einen Seite ist ein Anschluß für den Computer, auf der anderen Seite ein Stecker-

chen für das Telefon (bzw. die Telefonleitung).
Es gibt zwei unterschiedliche Stromzuständen beim Computer: Strom ein (ein BIT ist gesetzt), Strom aus (das BIT ist nicht gesetzt). Also brauchen wir auch zwei unterschiedliche Tonhöhen, um Daten senden und empfangen zu können. Ein sogenannter HALBDUPLEXBETRIEB wäre so schon möglich. In dieser Betriebsart sendet jeweils nur eine Station, während die andere Station nur Daten empfängt. Das hat allerdings Nachteile. Stellen Sie sich vor, sie rufen einen Riesentext ab, so ca. 20 Seiten lang, und stellen nach drei Zeilen fest, daß es doch der falsche Text war. Was tun Sie ? 20 Seiten lesen (schlecht, wenn Sie von Flensburg aus eine Datenbank in München anrufen! Das ist teuer), oder einfach den Telefonhörer auflegen (da freut sich der Betreiber der Datenbank, weil der Anschluß eine gewisse Zeit unnütz gesperrt ist!). Einem wirklichen Gesprächspartner würden Sie sagen: “STOP, die alte Story haste mir schon hundertmal erzählt”. Was liegt näher, als das der Datenbank mitzuteilen? Um einen VOLLDUPLEXBETRIEB zu gewährleisten, also einen Modus, in dem beide Seiten quasi gleichzeitig senden und empfangen können, benötigt man also ein weiteres Tonpaar. Man hat sich geeinigt, daß der Anrufer mit den beiden niedrigen Frequenzen arbeitet, während der Angerufene die beiden höheren Frequenzen nutzt. Die Norm, für die das folgende Bildchen gilt, nennt man CCITT V.21. Mit ihr kann man bis zu 300 Baud übertragen, das heißt, daß vom Sender zum Empfänger bis zu 300 Bits pro Sekunde verschickt werden können. Noch arbeiten nahezu alle Mailboxen der BRD mit dieser Übertragungsform.



Ein Akkustikkoppler bei der Kontaktaufnahme mit einem elegant geformten Telefonhörer.

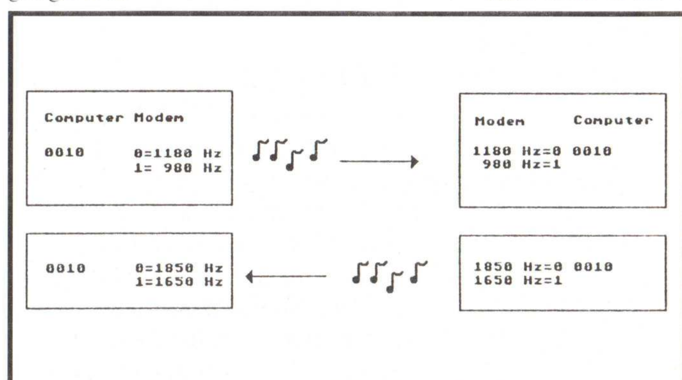
Das Ganze gibt ein schönes Bit-Geflüster. Wie sich so was anhört, kann sich auch ein Nicht-DFÜ'ler einmal anhören. Wählen Sie einfach 'mal die Telefonnummer 069/6638191. Am anderen Ende geht dann die Mailbox COMBO dran, das ist die Hausmailbox der Firma COMMODORE in Frankfurt. Bitte nur einmal ausprobieren, sonst steigt mir der Betreiber auf's Dach.
Bevor Sie aber jetzt in den Laden gehen, um sich ein Modem zu kaufen:

Problem 2:

Die Deutsche Bundespost mag MODEMs nicht so sehr. MODEMs werden nämlich DIREKT an die Telefonleitungen angeklemmt. Laut der Deutschen Bundespost könnten solche MODEMs den einwandfreien Ablauf des Fernmeldebetriebs stören! Entweder Sie mieten sich ein Postmodem (Geld ade), oder..

Lösung:

Kaufen Sie sich doch einen Akustikkoppler. Irgendein Computermensch kam auf die Idee: “Der Mensch wird ja auch nicht an die Telefonleitung angeklemmt, er SPRICHT in den Hörer, das kann der Computer doch auch!”. Und so konstruierte er sozusagen ein Negativ eines Telefonhörers. Es entstand ein Ding mit zwei Löchern, in das der Telefonhörer hereingelegt wird, (siehe Foto).
Schaltet man so einen Akustikkoppler an, wird man zumeist erst einmal angepfiffen. Übrigens muß ein solcher Akustikkoppler auch erst durch die Post geprüft werden, was den Hersteller Zeit und Geld kostet. Pech für den Verbraucher: Er muß statt DM 100,- Warenwert glatt das Drei- bis Zehnfache zahlen! Nach bestandener Prüfung gibt es die berüchtigte FTZ-Nummer (Fernmeldetechnische Zulassung), die sich auf dem Gerät befinden muß (Aufkleber/ Stempel). Ansonsten darf ein solches Gerät nicht betrieben werden!
Leider hat ein Akustikkoppler einen ganz gravierenden Nachteil: Er ist sehr störanfällig. Stößt man während der Übertragung mal zufällig an ihn dran, dann gibt es Datensalat auf dem Bildschirm. Bei Modems kann das nicht passieren, da sie ja direkt gekoppelt sind.
Eine weitere Möglichkeit gibt es erst seit kurzer Zeit: Die MODEMBOX. Für eine geringere Gebühr kann man sich durch die Post einen MODEMKERN in's Haus legen lassen (sozusagen ein dummes Modem). Erst durch die MODEMBOX, die man sich noch kaufen muß, wird das Modem betriebsbereit. Vorteil: Geringere monatliche Belastung, sichere Übertragung. Nachteil: Eine Modembox kostet zwischen 800,- und unbegrenzt!
Die grundlegenden Dinge der DFÜ haben Sie jetzt gelernt. Das war doch



Kommunikationsprinzip eines Modems.

nicht schwer, oder? Im nächsten Teil geht es in die technischen Details, bevor es dann endlich mit bundesdeutschen Mailboxen losgeht.

Zum Eingewöhnen hier schon einmal die Begrüßungsmeldung, das sogenannte Intro einer größeren Mailbox in Wiesbaden. Bis demnächst.

Ein Mailbox-Intro hat einiges zu bieten:

```
1{~S| carrier detected, connected to host...
 press return >

-OK-

V1.0 ( C )  by Dirk Graczik  ...........
V2.0        by   Hans  ............



    Titelbild mit CTRL-X abbrechbar

 SERVUS, GRUEZI, GUDE, TACH UND HALLO !
             HIER IST DIE
   ****
  **  **
  **  **                 \'''
W ******                `::::'
  **  **               `:():::'
I **  **               `:::::::'
                     \''':::::::'
E *****             \'''''::::''''
  **  **                 `.'''''
S **  **                 `.''''''
  *****                 `...''''''
B **  **                `...'''''''
  **  **  SYSOP    : HANS  `...'''rrr
A         COSYSOP  : ALKI  `...'''''''
   ****                   `...''''''
D **  **  PARAMETER : 8 N 1 `...''''''
  **  **                     `...'''''
E ******  ONLINE :            `..'''''
  **  **                       \'  \''
N **  **       24 Stunden      `    \'
                             \''    \'
   Box    06121 - 60 88 07            `
                                      `






Eingetragene Teilnehmer: 161

Sie sind der 8112. Anrufer am

Sonntag, den 14.02.1988 um 19:33 Uhr

- CHEF IST LEIDER NICHT DA -

Gaeste tragen sich bitte mit Gast ein !

Login    >LURCHI

Passwort >******

Moment bitte, bin am suchen...
```

```
- USERLEVEL 3-

Guten Abend LURCHI

Letzter Anruf am

Freitag, den 29.01.1988 um 23:48 Uhr

Sorry, ich habe keine Post fuer Dich.



Die A-R-A - Box begruesst LURCHI



Modul     : Hauptmenue
Uhrzeit   : 19:34
Bytes     : 00992
Restbytes: 25008
___________________
**        Hauptmenue              **
___________________
 00 Neueintraege        10 Inhalt
 20 Witzecke            30 Pers.Mail
 40 Programme           50 Pinboard
 60 A C C Information   70 An+Verkauf
 80 Userantraege        90 Usergroup *

 In Info              MO Kurzmenue E/A
 CO Sysop rufen       99 E N D E .....
___________________
 weitere hinweise ueber 10 Inhalt


LURCHI     >99



Auf Wiedersehen LURCHI

Dank fuer Deinen Anruf,

und schau bald mal wieder rein.



Dialog beendet am

Sonntag, den 14.02.1988 um 19:34 Uhr
Online - Zeit     > 00:01:39
Gelesene Bytes    > 001425



system update running...
```

VON ANDREAS KRÄMER

DIE ERFORSCHUNG VON BITS UND BYTES

DISCovery, der Diskettenmonitor

Disketten-Monitore sind ein Utility, um dessen Gebrauch die meisten AMIGA-Anwender früher oder später nicht mehr herumkommen. DISCovery ist ein solcher Disketten-Editor, der uns nach langer Reifezeit in einer endgültigen Version zum Test vorlag.
Der Monitor stammt von dem deutschen Programmierer Ralph Babel, mit dem wir des öfterern schon nähere Gespräche führen konnten.

Ein Disketten-Monitor? Wofür?

Diese Frage ist schnell beantwortet. Auch die beste Software ist nicht gegen Fehler hinsichtlich des Anwenders oder der Hardware geschützt, von Fehlern der Software wollen wir an dieser Stelle noch gar nicht sprechen. Genannt sei nur ein Programm, das auf jeder Workbench zu finden ist - DISK-DOCTOR. Leider verrichtet das Programm, das im Ordner "C" zu finden ist, seine Arbeit nicht immer zur Zufriedenheit des Users. Wichtige Dateien gehen des öfteren verloren, obwohl man gerade diese retten wollte.
Ein Disketten-Monitor sollte hinsichtlich vieler Anwendungen einige grundlegende Eigenschaften besitzen. Zum einen sollte er die Daten auf einer Diskette lesen bzw. wieder schreiben können, dies geschieht meistens, indem der Monitor sogenannte Tracks oder Sektoren einliest, die beim AMIGA einen festgelegten Umfang besitzen. Der Monitor sollte des weiteren die Möglichkeit besitzen, diese Daten zu modifizieren. In diesem Punkt unterscheiden sich die meisten Monitore voneinander, denn manche besitzen mehr, andere weniger Manipulationsmöglichkeiten. In erster Linie sollte aber ein Monitor zum Regenerieren von verlorenen Daten herangezogen werden.

Der Umfang

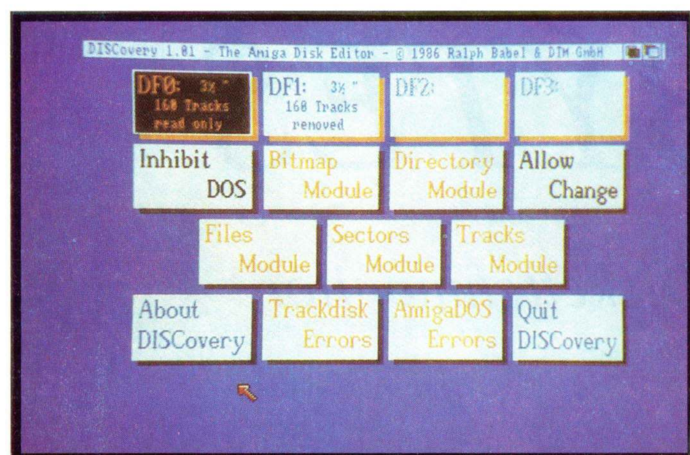
Im Lieferumfang von DISCovery befindet sich neben der Programm-Diskette ein etwa 160 Seiten starkes Handbuch. Um es gleich vorwegzunehmen: Würde jedes Handbuch eine solche Detailfülle bieten, wäre dem Anwender wahrlich gedient. Das Handbuch ist sowohl für den erfahrenen Anwender als auch für den Einsteiger bestens geeignet. Es erklärt einfach und verständlich Grundlagen über Datei-, Block- oder Directory-Organisation,

doch damit nicht genug, der Käufer findet bei genauen Durchlesen noch weit mehr Informationen über verschiedenste Themen.

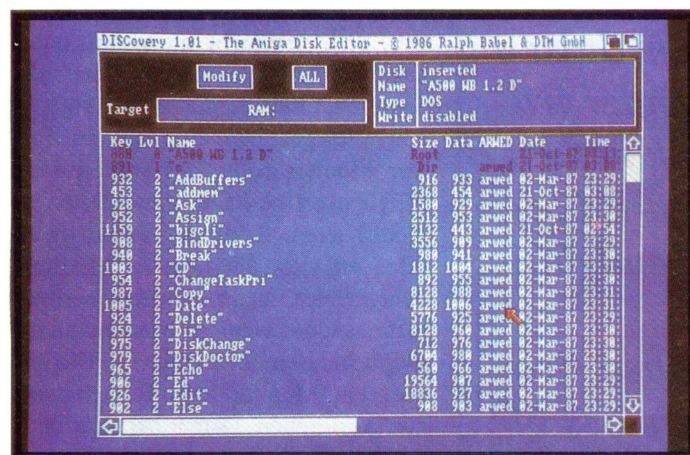
Ans Werk

Die DISCovery-Diskette wird anstelle der Workbench-Diskette eingelegt. Das System lädt daraufhin die Workbench. Nach Anklichen des Diskettensymbols öffnet sich ein Fenster, in dem sich neben den normalen Ordnern bzw. Dateien einer Workbench-Diskette das DISCovery-Piktogramm und ein SCA-Virus-Check-Programm finden. Dieses Bonus-Programm kann installiert werden und meldet dem Anwender beim Einlegen einer neuen Diskette, ob es sich um einen Standard-Bootblock handelt oder ob etwas nicht in Ordnung ist.

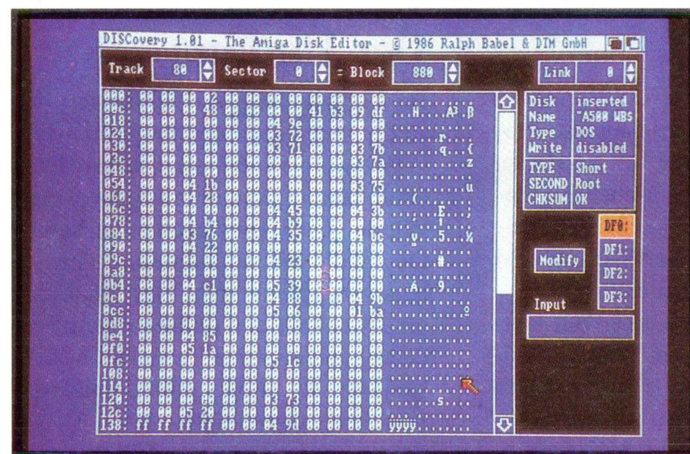
Durch Anwählen des DISCovery-Piktogramms erscheint zunächst ein Introbildschirm, der durch Drücken einer Taste verbannt werden muß. Ist dies erfolgt, so erscheint das Hauptmenü von DISCovery, das schnell die Komplexität des Programms erahnen läßt. Doch bevor man einen Menüpunkt anwählen kann, muß zum Manual gegriffen werden, denn das Programm verlangt die Eingabe eines Wortes auf einer zufällig ausgewählten Seite. Neben diesem mittlerweile sehr modern gewordenen Kopierschutz ist das Handbuch selbst gegen Kopieren geschützt, indem auf hellblauen Grund mit dunkelblauer Schrift gedruckt wurde. Das Kopiergerät bedankt sich mit einem einfarbigen Blatt. Über Sinn und Unsinn eines solchen Kopierschutzes läßt sich natürlich streiten, aber er erweist sich als immer noch besser als ein Kopierschutz auf der Diskette, bei dem nicht einmal eine Arbeitskopie angefertigt werden kann.



Das Hauptmenü von DISCovery.



Im Directory-Modul kann das Verzeichnis nach Belieben geändert werden.



Das Herzstück von DISCovery ist der Sector-Editor.

Beim Hauptmenü fällt zunächst die Anzeige der Laufwerke auf; nicht angeschlossene Laufwerke (beispielsweise df2: oder df3:) werden "ghosted" dagestellt. Des weiteren findet der Anwender 11 weitere Gadgets, die zum einen in die verschiedenen Unterprogramme verzweigen oder bestimmte Einstellungen festlegen. Um in ein Programm-Modul zu gelangen, gibt es insgesammt drei Möglichkeiten: zum einen kann das repräsentierende Gadget auf dem Bildschirm angewählt werden, zum zweiten kann dies auch über ein Pull-Down-Menue vonstatten gehen, und zusätzlich dazu geht es auch noch über Tastatursequenzen.

Das Bitmap-Modul

Dieses Modul erlaubt es dem Anwender, eine Diskette global zu überblicken. Er kann schnell und einfach beliebige Blöcke modifizieren. So kann er beispielsweise als belegt geltende Blöcke als frei deklarieren. Natürlich schreibt das Programm die vorgenommenen Änderungen nur nach einem entsprechenden Befehl zurück auf Diskette. Über Pull-Down-Menues, Gadgets und Tastatursequenzen kann der Anwender die gewünschten Schritte einleiten.

Das Directory-Modul

Dieses Modul ist schon weit interessanter als das Bitmap-Modul. Es zeigt die globale Directory mit allen Subdirectories und Files auf. Zusätzlich gibt das Programm dem Anwender Infor-

mationen, in welchen Block der Eintrag zu finden ist, in welcher Verschachtelungstiefe sich die Directory bzw. die Datei befindet. (Das Anlegen eines globalen Verzeichnisses einer AMIGA-Diskette kann in einer Baumstruktur erfolgen, in einem Directory kann ein Subdirectory usw. angelegt werden, wodurch sich eine Baumstruktur ergibt. Die Verschachtelungstiefe gibt darüber Auskunft, in welcher Ebene sich die jeweilige Datei befindet.) Man erhält auch Informationen über den Namen, die Größe, die Adresse des ersten Datenblocks, den Disketten-Status, über Erstellungsdatum bzw. -zeit und einen vorhandenen Kommentar. Dem Anwender steht hier eine Vielzahl von Möglichkeiten offen, um am Verzeichnis herumzuwerkeln.

Das Files-Modul

Mit diesem Modul steht dem Anwender ein File-Monitor zur Verfügung, der jedoch sehr eingeschränkt ist. Das Modifizieren einer ausgewählten Datei geschieht recht umständlich, denn keine ASCII-Zeichen können direkt eingegeben werden; der jeweilige Kode muß für jedes Zeichen nachgeschlagen werden. Dieses Manko ist leider in allen Modulen zugegen. Ein solches Modul ist aber zweifelsohne eine Zugabe für einen Disketten-Monitor. Trotzdem steht dem Anwender eine Reihe von Optionen zur Verfügung. Verschiedene Ausgabearten können gewählt oder es kann nach bestimmten Sequenzen gesucht werden. Dieses Modul erwies sich als recht hilfreich.

Das Sector-Modul...

...ist das Herzstück von DISCovery. Es beschäftigt sich mit dem Edieren von Blöcken. Leider erfolgen die Änderungen über den sehr umständlichen Weg, der schon beim Files-Modul beschrieben wurde. Hier jedoch wiegt dieses Manko schon schwerer. Ansonsten wird fast alles geboten, was von einem Disketten-Monitor verlangt werden kann. Besondere Aufmerksamkeit fand das "Misc."-Menue. Hier kann beispielsweise der Hashwert von bis zu 30 Zeichen berechnet oder die Prüfsumme neu erstellt werden. Aber auch Such-Funktionen nach bestimmten Zeichenketten stehen dem Anwender zur Verfügung.

Das Tracks-Modul

Dieses Modul vereinigt eine Sammlung von Befehlen, die das Bearbeiten von globalen Tracks gestattet. Das Besondere ist hierbei, daß Fehler leicht erkannt und eventuell beseitigt werden können. Kann ein Track nicht auf dem normalen Weg eingelesen werden, steht eine bestimmte Leseroutine zur Verfügung. Der fehlerhafte Track kann daraufhin neu formatiert und der im Speicher befindliche Track zurückgeschrieben werden. Das Programm unterscheidet jeweils zwischen Trackdisk- und AMIGADOS-Fehlern. Um welchen Fehlern es sich handelt, kann mit zwei weiteren Modulen festgestellt werden.

Fazit

DISCovery genügt auch höheren Ansprüchen. Die vielen Möglichkeiten, die das Programm bietet, sind für einen Disketten-Monitor nicht üblich. Das Programm ist sowohl für den Laien als auch für den Profi geeignet. Als Manko muß jedoch die umständliche Eingabe der vom Anwender gewünschten Änderungen genannt werden.

Da jeder Anwender um die Benutzung eines Disketten-Monitors früher oder später doch nicht herumkommt, kann zum Kauf von DISCovery geraten werden. Auch das sehr gute Handbuch sollte in die Kaufentscheidung mit einbezogen werden.

DISCOVERY

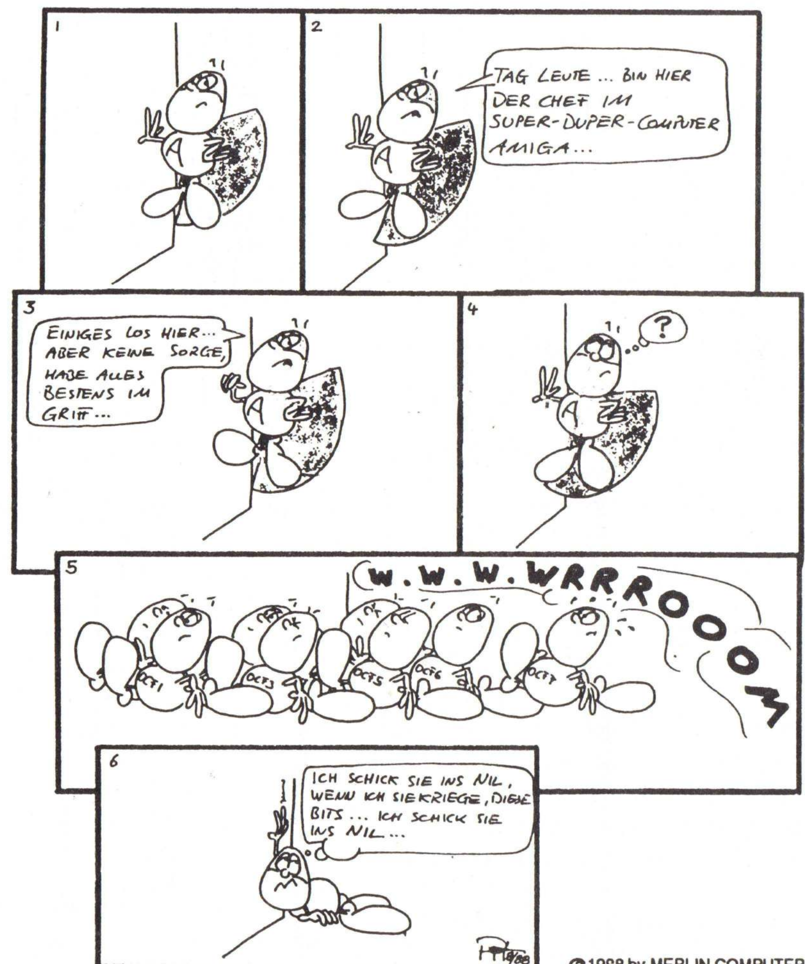
+ gutes Handbuch
+ Bedienung per Maus (Gadgets, Pull-DownMenüs) und Tastatur
+ komfortable Fehleranalyse
+ kein Kopierschutz
- umständliches Ändern von Daten

Anbieter:

DTM
Poststr. 25
6200 Wiesbaden-Bierstadt
Tel. 06121-560084
Preis: 198.- DM

ENDE

Liebe Leser,
als ich im Herbst ´86 meinen PC von Big Blue gegen einen AMIGA in Zahlung gab, ahnte ich noch nichts von IHM.
Auch später - der gute alte 1000er war inzwischen mit Mega und Zweitlaufwerk ausgestattet - wollte ich IHN nicht wahr haben.
Heute jedoch kann es keinen Zweifel mehr geben: ER existiert! ER: Der Superheld im Systembus, der Übernatürliche, der Stählerne, der unumschränkte Herrscher im AMIGA. ER:
AMIGA-Man!



Die KICKSTART Redaktion hat es sich nun zur Aufgabe gemacht, die Welt über diese bahnbrechende Entdeckung zu informieren.
Lesen Sie deshalb heute und künftig in KICKSTART: AMIGA-Man....Abenteuer im AMIGA.
Ihr Pit Burkhardt

VON B. VAN RÜSCHEN

BOOTLETTER

Der eigene Bootvorspann

Bootet man eine Diskette, z.B. die Workbench, dann fällt dem Amiga meistens nichts besseres ein, als bis zum Erscheinen des CLI-Fensters einen leeren, weißen Bildschirm darzustellen. Wäre es nicht besser, wenn in dieser Zeit schon ein informativer Text auf dem Bildschirm erscheinen würde? Dies ermöglicht das Programm "BootLetter" und macht da weiter, wo andere aufhören.

Den Ansatz dafür bietet natürlich der "Boot-Block". So bezeichnet man die ersten zwei Blöcke einer Diskette, die ein Programm enthalten, das beim Booten als erstes die DOS-Library initialisiert. Da dieses Programm aber nur wenige Bytes lang ist, aber insgesamt 1012 Bytes dafür zur Verfügung stehen, bieten sich Erweiterungen an. "BootLetter" installiert auf dem Boot-Block ein kurzes, aber flexibles Programm, das auf dem Bildschirm einen Text mit 40 Zeichen pro Zeile und 15 Zeilen darstellt. Das kann sogar mit vertikalem Scrolling geschehen. Es werden zusätzlich Farb-Abstufungen auf dem Hintergrund oder der Textfarbe eingerichtet. Der Boot-Text schaltet sich entweder durch Erscheinen des CLI-Fensters ab, oder ist auch während der Ausführung der Startup-Sequence zu sehen und wird dann über einen bestimmten CLI-Befehl abgeschaltet. Boot-Programm und Text nehmen den Boot-Block also bis auf das letzte Byte in Anspruch.

Nun zum "BootLetter" selbst: Dieser präsentiert sich auf der Workbench durch ein kompaktes Fenster mit vielen Schaltern und Reglern, den "Gadgets". Diese Gadgets können Sie mit der Maus anklicken und damit folgende Parameter einstellen:

Farbeinstellungen

Mit den rechten und linken Schiebereglern können Sie die Rot-, Grün- und Blau-Anteile der Text- und Hintergrundfarbe, die das Boot-Programm verwenden soll, einstellen. Entsprechend der Einstellung werden auch die Workbench-Farben geändert. Das Gadget mit der Beschriftung "<¡>" vertauscht die Farbeinstellungen. Mit dem Gadget unterhalb der Beschriftung "Abstufung auf" können Sie auswählen, ob Hintergrund oder Text in 16 Farbabstufungen (Farbtönen) dargestellt werden. Wie diese Abstufungen verlaufen, können Sie mit den eingerahmten RGB-Gadgets einstellen. Dabei wird ein angewähltes RGB-Gadget farblich hervorgehoben. Entsprechend den angewählten Gadgets werden dann später bei der ausgewählten Farbe die RGB-Anteile auf den Höchstwert gesetzt und bei jeder neuen Abstufung verringert. Wenn Sie das "R" anwählen, wird der Rot-Anteil Ihrer Farbe verringert. Wenn Sie alle Anteile anwählen, würden nur Grau-Abstufungen erscheinen. Sind keine Anteile angewählt, dann erscheinen auch keine Farbabstufungen, es wird nur die eingestellte Farbe dargestellt.

Sonstige Einstellungen

Einen ganz wichtigen Parameter stellt auch das Abschaltungs-Gadget dar. Zeigt das Gadget "auto" an, dann verschwindet der vom Boot-Programm ausgegebene Text genau dann vom Bildschirm, wenn das CLI-Fenster erscheint. Wenn das Gadget "CLI" anzeigt, müssen Sie den Boot-Text über einen CLI-Befehl abschalten. Dieser Befehl lautet "BootOff" und wird beim Installieren des Boot-Blocks in die C-Directory Ihrer Diskette geschrieben. Mit dem Scrolling-Gadget können Sie einstellen, wie der Boot-Text erscheinen soll. Wenn das Gadget "ein" anzeigt, dann scrollt der Boot-Text solange vertikal, bis die erste Zeile des Textes den oberen Bildschirmrand erreicht hat. Wird "aus" angezeigt, dann wird nicht gescrollt, und der Text ist gleich voll sichtbar.

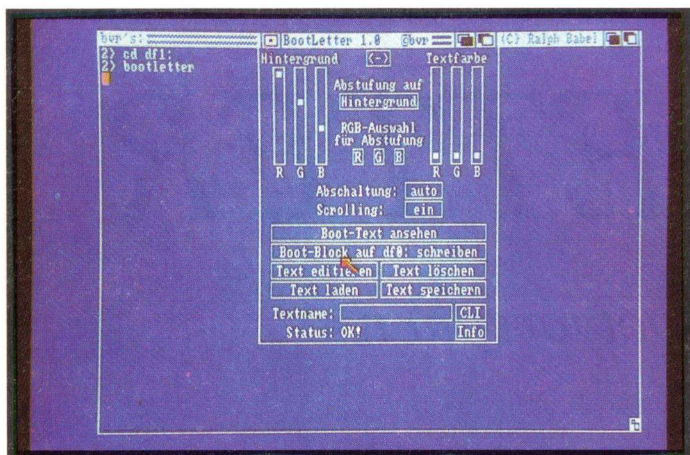


Bild 1: Nach Starten des Programms öffnet sich ein Fenster mit den vielen Einstellungsmöglichkeiten.

Nun zu den eigentlichen Funktionen des "BootLetters":

1. Boot-Text ansehen

Diese Funktion zeigt den Boot-Text so an, wie er auch später beim Booten erscheint, also mit sämtlichen Einstellungen. Zum "BootLetter" kommen Sie durch Mausklick oder einen sonstigen Tastendruck zurück.

2. Boot-Block auf df0: schreiben

Das Boot-Programm wird auf den Boot-Block der Diskette im internen Laufwerk geschrieben. An dieser Stelle eine Warnung: Sie sollten keine Originale mit dieser Funktion bearbeiten. Es gibt Programme, die den Boot-Block als Kopierschutz verwenden oder eigene Lade-Routinen dort installiert haben. Wenn das Abschaltungs-Gadget auf "CLI" steht, dann wird zusätzlich noch das Programm "BootOff" in die C-Directory Ihrer Diskette geschrieben.

3. Text edieren

Wenn Sie dieses Gadget anklicken, werden Sie Bekanntschaft mit dem guten alten "Ed" machen. Das Ed-Fenster wurde auf die für den Boot-Text nötigen Maße verkleinert, nämlich auf 15*40 Zeichen. Daß tatsächlich 41 Zeichen pro Zeile angezeigt werden, liegt daran, daß sonst das Word-Wrapping auf 40 Zeichen nicht funktioniert. Ansonsten hat sich nichts geändert. Wenn Sie Ihren Text eingegeben oder geändert haben, dann können Sie den Editor wie immer mit der Tastenfolge "Esc", "x" und "Return" verlassen.

4. Text löschen

Klicken Sie einfach dieses Gadget an, wenn Sie den aktuellen Text löschen wollen, um danach einen neuen einzugeben. Die Parameter bleiben dabei erhalten. Erschreckt Sie die Sicherheitsabfrage? Nun, ein Requester hätte das Programm nur unnötig länger werden lassen, und der Alert erfüllt ja auch seinen Zweck. Die Abfrage erscheint übrigens nicht, wenn der Text bereits gelöscht oder ohne Änderung der Parameter gespeichert wurde.

5. Text laden

Um einen Text zu laden, müssen Sie vorher das Textname-Gadget anklikken. Sie können nun den Namen Ihres Textes mit max. 40 Zeichen eingeben. Wenn Sie dann "Text laden" anklikken, wird Ihr Text geladen. Zusätzlich wird noch ein Parameter-File geladen, das Farben und sonstige Einstellungen enthält.

6. Text speichern

Nachdem Sie den Textnamen eingegeben haben, klicken Sie dieses Gadget an. Ihr Text wird nun im normalen ASCII-Format gespeichert. Zusätzlich wird auf Ihrer Diskette ein File geschrieben, das die aktuellen Einstellungen enthält. Dieses File trägt den gleichen Namen wie Ihr Text, nur mit dem Unterschied, daß an den Namen noch ein "&" gehängt wird.

7. CLI

Gleich neben dem Rechteck, das zur Eingabe Ihres Textnamens dient, befindet sich das "CLI"-Gadget. Nach dem Anklicken wird ein neues CLI-Fenster geöffnet, wobei das "BootLetter"-Programm im Multitasking-Betrieb weiterläuft. Wenn Sie das Gadget mehrmals anklicken, dann werden auch mehrere CLI-Fenster geöffnet.

8. Info

Dann hätten wir da noch das Info-Gadget. Nach dem Anklicken erscheint ein Alert, der Sie kurz über den edlen Autor informiert. Um das Programm zu beenden, sollten Sie das Schließ-Gadget in der linken oberen Ecke des Windows anklicken. Haben Sie Ihren Text mit den eingestellten Parametern nicht gespeichert, erscheint wie bei "Text löschen" eine Sicherheitsabfrage. Drücken Sie jetzt die rechte Maustaste für "Abbruch", wird das Beenden des Programms verhindert.
Das waren also alle Funktionen des "BootLetters". Es bleiben aber noch einige erwähnenswerte Punkte:

Die Statusanzeige

Über Fehler z.B. bei den Disketten-Zugriffen werden Sie in der Statusanzeige informiert. Normalerweise wird aber "OK" ausgegeben. In dieser Anzeige erscheinen auch Informationen über den Verlauf bestimmter Funktionen. So informiert Sie das Programm beispielsweise, wenn der CLI-Befehl "BootOff" auf Ihre Diskette geschrieben wird. Wundern Sie sich nicht über die Fehlermeldung "Schnipp!". Diese erscheint immer dann, wenn eine Zeile des Boot-Textes über 40 Zeichen lang ist. Der Rest der Zeile wird dann abgeschnitten.

"BootOff"

Wenn das Abschaltungs-Gadget "CLI" anzeigt, dann wird beim Installieren des Boot-Blocks zusätzlich noch der CLI-Befehl "BootOff" in die C-Directory Ihrer Diskette geschrieben. Booten Sie diese Diskette, dann wird sich der Boot-Text nicht automatisch abschalten. Wenn das System die Startup-Sequence abarbeitet, sollten Sie einmal "Ctrl" + "d" drücken. Geben Sie nun blind "BootOff" ein. Sie müssen keine weiteren Parameter eingeben. Der Boot-Text verschwindet, und es erscheint das CLI-Fenster. Nun haben Sie Gelegenheit, diesen neuen Befehl in die Startup-Sequence einzubauen. Und so arbeitet der Befehl: Mit



Bild 2: Ein Beispiel von BOOTLETTER.

dem Boot-Text wird auch eine Interrupt-Routine installiert. "BootOff" löscht einen Sprung-Befehl dieser Routine und bewirkt damit, daß die Routine den Boot-Text und sich selbst abmeldet. "BOOTOFF" ist ein kleines Assembler-Programm, das Sie am Ende des Artikels finden.

Die Listings

Sie finden "BOOTLETTER" in zwei Listing-Versionen. Einmal liegt das Programm als BASIC-Hexdump vor. Damit ist es für alle Amiga-Besitzer zum Abtippen geeignet. Sie sollten das BASIC-Programm aber mit anderem Namen abspeichern, denn nach dem Start wird das eigentliche Assembler-Programm auf Ihre Diskette geschrieben. Das Programm hat eine Länge von über 4200 Bytes. Das ist aber noch recht wenig, wenn man bedenkt, daß z.B. die Strukturen für die Gadgets in einem kompaktem Format abgelegt sind (3 Bytes pro Struktur statt 80). Sie können das Programm vom CLI aus durch Eingabe von "BootLetter" und Drücken von "RETURN" starten. Des weiteren wollen wir unseren Lesern den eigentlichen Source-Code nicht vorenthalten, so daß Sie Einblicke in die eigentliche Programmierung des einen oder anderen Features erhalten. Für viele ist dies sicherlich hochinteressant. Verwendet wurde der Profimat-Assembler von Data-Becker.

Wichtig ist noch folgendes: Der "BootLetter" benötigt die Files "Ed", "NewCli" und "Run" aus der C-Directory der Workbench. Achten Sie also darauf, daß sich diese Files auch auf der entsprechenden Diskette befinden. Ansonsten informiert Sie das Programm, wenn es Probleme mit "Ed" gibt.

Zum Schluß noch ein paar Tips: Der "BootLetter" eignet sich auch als Virus-Killer. Sollte Ihr Text beim Booten nicht mehr erscheinen, dann können Sie davon ausgehen, daß sich ein "Virus" auf Ihrer Diskette niedergelassen hat. Solche Viren haben die unangenehme Eigenschaft, sich selbst nach einem Reset auf den Boot-Block der eingelegten Diskette zu kopieren. Sie sollten dann Ihren Boot-Text noch einmal installieren, wodurch der Virus gelöscht wird. Der Boot-Block läßt sich aber auch anders nutzen: Man kann dort auch ein Programm ablegen, das die restlichen Blöcke des ersten Tracks einliest. Diesen hat Ihr Amiga sowieso komplett im Speicher, da immer auf einen ganzen Track zugegriffen wird. Der Vorteil daran ist, daß z.B. eine Grafik, die den ganzen Bildschirm ausfüllt, erscheinen und nebenbei noch digitalisierter Sound erklingen kann. Der Nachteil dieser Methode: 10 KB sind auf Ihrer Diskette belegt. Der "BootLetter" gibt sich ganz ohne Nachladen nur mit dem Boot-Block zufrieden, wird dadurch aber nicht weniger wirkungsvoll! Natürlich finden Sie das Programm auch auf unserer Monats-Diskette, so daß Sie sich die Tipparbeit sparen können. Auf der Diskette finden Sie alle Versionen des Programms - Hexdump-Listing, Assembler-Listing und die lauffähige Version.

Die Redaktion wünscht Ihnen viel Spaß mit "BOOTLETTER".

```
;BootOff   (c)KICKSTART

move #$4e71,$128 ;NOP setzen =
                 keine Verzweigung
Wait:
 cmp.l #$100,$6c  ;Warten, bis
                 Interrupt-Routine
 beq.s Wait        ;sich
                  abgeschaltet hat
 rts
 END
```

Das Programm BOOTOFF, schaltet den Boottext aus dem CLI wieder aus.

```
; BootLetter V 1.0
; Assembler-Quelltext
;     (PROFIMAT)
;  (c) KICKSTART 1988
;### MACROS ###

_LVOForbid   = -132              ;Offsets für
_LVOFindTask = -294              ;Systemroutinen
_LVOGetMsg   = -372
_LVOReplyMsg = -378
_LVOWaitPort = -384

 PR_CLI     = 172
PR_MsgPort = 92
 ExecBase   = 4

 CALL:MACRO $\1                 ;CALL  Systemroutine
  jsr _LVO\1(a6)                    ;—  aufrufen
  ENDM

 INIT_AMIGA:MACRO                 ;INIT_AMIGA
  move.l ExecBase,a6              ;———
   sub.l a1,a1                    ;eigenen Task holen
  CALL FindTask
   move.l d0,MyTask             ;und Pointer ablegen
   move.l d0,a4
  tst.l PR_CLI(a4)
   bne.s \1
   lea PR_MsgPort(a4),a0  ;Start von Workbench
   CALL WaitPort
  lea PR_MsgPort(a4),a0         ;Startup-Message
   CALL GetMsg                   ;holen
 move.l d0,WBenchMsg
 \1:                             ;Start von CLI
  ENDM

  EXIT_AMIGA:MACRO                    ;EXIT_AMIGA
   tst.l WBenchMsg                    ;———
```

```
   beq.s \1                    ;bei Start
  CALL Forbid                ;von Workbench
  move.l WBenchMsg,a1        ;Message abmelden
  CALL ReplyMsg
 \1:
   moveq #0,d0               ;Return-Code löschen
   rts
  ENDM

  BSS
 WBenchMsg: DC.L 0
 MyTask:    DC.L 0
  TEXT


 ;### HAUPTPROGRAMM ###

 _LVOFindResident   = -96      ;Offsets für
 _LVODisable        = -120     ;Systemroutinen
 _LVOEnable         = -126
 _LVOAllocMem       = -198
 _LVOFreeMem        = -210
 _LVOAddPort        = -354
 _LVORemPort        = -360
 _LVOOldOpenLibrary = -408
 _LVOCloseLibrary   = -414
 _LVOOpenDevice     = -444
 _LVOCloseDevice    = -450
 _LVODoIO           = -456
 _LVOOpen           = -30
 _LVOClose          = -36
 _LVORead           = -42
 _LVOWrite          = -48
 _LVODeleteFile     = -72
 _LVOExecute        = -222
 _LVOText           = -60
 _LVOOpenFont       = -72
 _LVOMove           = -240
 _LVOSetRGB4        = -288
 _LVOSetAPen        = -342
 _LVOSetDrMd        = -354
 _LVOCloseWindow    = -72
 _LVODisplayAlert   = -90
 _LVOOpenWindow     = -204
 _LVORectFill       = -306
 _LVORefreshGList   = -432

 MEMF_CHIP     = 1<<1          ;Definitionen
 MEMF_CLEAR    = 1<<16
 MODE_OLDFILE  = 1005
 MODE_NEWFILE  = 1006
 CMD_WRITE     = 3
 CMD_UPDATE    = 4
 TD_MOTOR      = 9
 TD_PROTSTATUS = 15
 RP_COMPLEMENT = 2
 ScreenSize    = 40*17*16
 GadgetSize    = 6*74+16*80+36+42
 BfSize        = 20000
 FileName   EQUR Gadgets+GadgetSize-42
 BootTxName EQUR RunEd+7

  jmp Startup(pc)               ;Libs,Ed,Window,
                                ;Gadgets,Text

 ;                       ===> HAUPTSCHLEIFE
 Main:
  lea Gadgets,a5                ;Prop-Gadgets
  moveq #1,d7
 SetCol:
   asl.l #8,d6
   move.l WbViewPort(pc),a0
   move d7,d0
   eor #1,d0
   move.b 48(a5),d1              ;Werte der
   lsr.b #4,d1                   ;Prop-Gadgets
   or.b d1,d6                    ;lesen und
   asl.l #4,d6                   ;zu Farbwerten
  move.b 74+48(a5),d2            ;umwandeln
   lsr.b #4,d2
   or.b d2,d6
   asl.l #4,d6
  move.b 148+48(a5),d3
   lsr.b #4,d3
   or.b d3,d6
  move.l GfxBase(pc),a6
   CALL SetRGB4                  ;Farben setzen
  add #222,a5
  dbra d7,SetCol
  move.l d6,ActualCol            ;Farben sichern
  btst #0,ParFlags(pc)
  bne.s StoreCol
  swap d6
 StoreCol:                       ;Farben umformen
  move ColMask(pc),d0            ;und in
  or d0,d6                       ;Interrupt-Routine
  move d6,ColValue1+2            ;schreiben
  swap d6
  move d6,ColValue2
  and #$111,d0                   ;Abstufungs-Wert
  move d0,ColSub+2               ;setzen

  move.l MyWindow(pc),a0         ;IntuiMessage
  move.l 86(a0),a0               ;holen und
  move.l ExecBase,a6             ;auswerten
  CALL GetMsg
  tst.l d0
  beq Main
  move.l d0,a1
  move.l d0,a5                   ;Message abmelden
  CALL ReplyMsg
  move 22(a5),d0                 ;IDCMP-Flags
  cmp.l #$100,$6c                ;Interrupt-Routine
   bne.s AskFlag                 ;angeschaltet?
  move #$4e71,IntOnOff-IntPrg+$100;Ja,dann aus
   clr.l d0                               ;$4e71 =
NOP
 AskFlag:
  cmp #$200,d0                   ;Close-Gadget?
  beq ExitMain
  cmp #$40,d0                    ;sonstiges Gadget
  bne Main                       ;angewählt?
  move.l 28(a5),a5               ;Ja, dann Gadget-
  move 38(a5),d0                 ;Nummer holen
  cmp #15,d0                     ;Parameter oder
Text
   bcc.s NoChange                ;geändert?
   clr.b SaveFlag                ;Ja, merken
 NoChange:
   clr.l Error                  ;Fehler-Var. löschen
   move.b ParFlags(pc),OldFlags;Gadget-Flags
                                  retten

 ;                       ===> GADGETS ABFRAGEN

   cmp #6,d0                      ;R-Gadget
                                  angeklickt?
  bne.s Gadget2                  ;Ja, dann Maske
für
  eor #$f00,ColMask              ;Abstufung ändern
 Gadget2:
   cmp #7,d0                 ;G-Gadget angeklickt?
   bne.s Gadget3                ;Ja, dann Maske für
   eor #$0f0,ColMask            ;Abstufung ändern
 Gadget3:
   cmp #8,d0                 ;B-Gadget angeklickt?
   bne.s Gadget4                ;Ja, dann Maske für
   eor #$00f,ColMask            ;Abstufung ändern
```

```
Gadget4:
 cmp #9,d0                        ;"Abstufung auf"
 bne.s Gadget5                    ;angeklickt?
 bchg #0,ParFlags                 ;Ja, merken
Gadget5:
  cmp #10,d0                    ;Abschaltungs-Gadget
  bne.s Gadget6                     ;angeklickt?
 bchg #1,ParFlags                 ;Ja, merken
Gadget6:
 cmp #11,d0                       ;Scrolling-Gadget
 bne.s Gadget7                    ;angeklickt?
 bchg #7,ParFlags                 ;Ja, merken

Gadget7:
 cmp #12,d0                       ;"<->" angeklickt?
 bne.s Gadget8
 move.l ActualCol(pc),d6          ;Aktuelle Farbe
 jsr SetProp2(pc)                 ;Props vertauschen
 jsr RefreshProp(pc)              ;und refreshen
 clr.l d0

Gadget8:
 cmp #15,d0                  ;"Boot-Text ansehen"
 bne.s Gadget9               ;angeklickt?
 jsr TestTx(pc)              ;Text vorhanden?
 bne.s Gadget9               ;Nein, verzweigen!
lea IntOnOff(pc),a5          ;Abschaltungs-Befehl
                                  retten
  move.b (a5),d7             ;"bra"($60) = Boot-Text
                                  ein
 move.b #$60,(a5)              ;Abschaltung dann mit
                                  "nop"
  jsr BootPrg(pc)                 ;Boot-Programm
                                   starten
 move.b d7,(a5)              ;Befehl zurücksetzen

Gadget9:
  cmp #16,d0                         ;"BootBlock auf
                                      df0:"
 bne Gadget10                     ;angeklickt?
 jsr TestTx(pc)                   ;Text vorhanden?
 bne Gadget10
 lea BootStat1(pc),a4
 jsr PrintStat(pc)                ;Status ausgeben
 clr.l ChkSum
 clr.l d0
 move #255,d1
 lea BootBlock(pc),a0             ;Checksumme
MakeChkSum:                       ;berechnen...
 add.l (a0)+,d0
 bcc.s NoSumAdd
 addq.l #1,d0
NoSumAdd:
 dbra d1,MakeChkSum
 moveq #-1,d1
 sub.l d0,d1
 move.l d1,ChkSum                 ;...und ablegen
 lea ReplyPort,a1
 move.l MyTask,16(a1)              ;Port
 CALL AddPort                     ;installieren
 lea IOExtTD,a1
 moveq #0,d0                       ;df0:
 clr.l d1
 lea TD_NAME(pc),a0               ;Trackdisk-Device
  CALL OpenDevice                      ;öffnen
  tst.l d0                             ;Fehler?
  bne SetTDErr                         ;Dann
verzweigen!
  lea IOExtTD,a1
  move.l #ReplyPort,14(a1)         ;Schreibschutz
  move #TD_PROTSTATUS,28(a1)       ;testen...
  clr.l 36(a1)
  CALL DoIO
```

```
 lea IOExtTD,a1
 moveq #28,d6
 tst.l 32(a1)                          ;gesetzt?
 bne.s MotorOff                   ;Dann verzweigen!
 move #CMD_WRITE,28(a1)
 move.l #BootBlock,40(a1)
 move.l #2*512,36(a1)                  ;BootBlock in
 clr.l 44(a1)                     ;Trackdisk-Buffer
 CALL DoIO                             ;schreiben
 lea IOExtTD,a1
 move.b 31(a1),d6                      ;Fehler?
 bne.s MotorOff                   ;Dann verzweigen!
 move #CMD_UPDATE,28(a1)
 clr.l 36(a1)                          ;BootBlock auf
 CALL DoIO                             ;Disk schreiben
 lea IOExtTD,a1
 move.b 31(a1),d6
MotorOff:
 move #TD_MOTOR,28(a1)                 ;Motor aus
 clr.l 36(a1)
 CALL DoIO
 lea ReplyPort,a1
 CALL RemPort                          ;Port löschen
 lea IOExtTD,a1
 CALL CloseDevice                  ;Device schließen
 tst.b d6                       ;Fehler vorgefallen?
 beq.s NoTDError
 move.l #WriteProt,Error               ;Schreibschutz
 cmp.b #28,d6                          ;war gesetzt
 beq SetGad
SetTDErr:
 move.l #DiskErr,Error             ;sonstiger Fehler
 bra SetGad
NoTDError:                             ;Muß CLI-Befehl
 btst #1,ParFlags(pc)                ;zur Abschaltung
 beq SetOK                      ;geschrieben werden?
 lea BootStat2(pc),a4
 jsr PrintStat(pc)              ;Ja, Status ausgeben
 move.l #BootOffName,d1
 lea BootOff(pc),a5                    ;"BootOff" auf
 moveq #60,d6                          ;Disk schreiben
 jsr CopyMemToFile(pc)
 beq SetOK                             ;Fehler?

Gadget10:
 cmp #13,d0                        ;"Text editieren"
 bne.s Gadget11                        ;angeklickt?
 jsr ClrStat(pc)                ;Statuszeile löschen
 lea RunEd(pc),a5
 jsr MyEx(pc)                          ;Ed aufrufen
 beq.s SetBootTx                   ;Boot-Text setzen

Gadget11:
 cmp #14,d0                            ;"Text löschen"
 bne.s Gadget12                        ;angeklickt?
 jsr Sure(pc)
                                 ;Sicherheitsabfrage
 beq.s Gadget12
 jsr DeleteTx(pc)                      ;Text löschen
 st.b TxClrFlag                        ;Kein Text
                                         vorhanden!
 bra SetOK                             ;OK ausgeben

Gadget12:
 cmp #17,d0                            ;"Text laden"
 bne Gadget13                          ;angeklickt?
 jsr ClrStat(pc)                ;Statuszeile löschen
 move.l #FileName,d1
 move.l #BootTxName,d7                 ;File ins RAM
 jsr FileCopy(pc)                      ;kopieren
 bne Gadget13                          ;Fehler?
 jsr SetParName(pc)
 jsr CopyFileToMem(pc) ;Parameter-File laden
 clr.b (a4)                            ;alter Filename
```

```
    move.l ActualCol(pc),d6
    jsr SetProp(pc)                 ;Props setzen
    jsr RefreshProp(pc)             ;und refreshen
    st.b SaveFlag                   ;keine Text-Änderung
  SetBootTx:
    move.l #BootTxName,d1
    jsr OpenOld(pc)                 ;BootText öffnen
    clr.l Error
    tst.l d0
    beq.s Gadget13                  ;Fehler?
    clr.b TxClrFlag                 ;Text vorhanden!
    move.l d0,d7
    lea BootTx(pc),a5               ;BootText Startadr.
    move #600-1,d0
  ClrBootTx:
    clr.b (a5,d0)                   ;BootText löschen
    dbra d0,ClrBootTx
  ReadTx:
    moveq #0,d6                     ;Zeichen-Pos.in Zeile
  ReadTxLoop:
    move.l d7,d1
    move.l #CharBf,d2
    moveq #1,d3
    CALL Read                       ;1 Zeichen lesen
    tst.l d0                        ;File-Ende?
    beq.s ExitRead                  ;Dann verzweigen!
    not.l d0                        ;Fehler?
    beq.s ExitRead                  ;Dann verzweigen!
    move.b CharBf(pc),d1            ;Zeichen auswerten
    cmp #$a,d1                      ;CR? Dann neue Zeile
    beq.s NextLine
    addq.b #1,d6
    bmi.s NextLine                  ;Kein Zeilenende?
    cmp #41,d6                      ;unter 41 Zeichen?
    bcs.s StoreChar                 ;Ja, dann ablegen
    move.l #CutTx,Error             ;Nein, "Schnipp!"
    bra.s ReadTxLoop                ;nicht ablegen
  StoreChar:
    sub.b #32,d1                    ;Zeichencode
    bpl.s CharOK                    ;umformen
    sub.b #32,d1
  CharOK:
    move.b d1,-1(a5,d6)             ;Zeichen ablegen
    bra.s ReadTxLoop
  NextLine:
    add #40,a5                      ;Neue Zeile
    cmp.l #BootTx+600,a5            ;BootText-Ende
    bne.s ReadTx                    ;erreicht?
  ExitRead:
    move.l #"(c)K",BootBlock+1020
    jsr MyClose(pc)                 ;Schließen
    beq.s SetOK                     ;OK ausgebn

  Gadget13:
    cmp #18,d0                      ;"Text speichern"
    bne.s Gadget14                  ;angeklickt?
    jsr TestTx(pc)                  ;Text vorhanden?
    bne.s Gadget14
    jsr ClrStat(pc)                 ;Statuszeile löschen
    move.l #BootTxName,d1
    move.l #FileName,d7             ;File auf Disk
    jsr FileCopy(pc)                ;speichern
    bne.s Gadget14                  ;Fehler?
    jsr SetParName(pc)              ;Parameter-File
    jsr CopyMemToFile(pc)           ;speichern
    clr.b (a4)                      ;alter File-Name
    tst.l Error                     ;Fehler?
    bne.s Gadget14                  ;Ja, verzweigen
    st.b SaveFlag                   ;keine Text-Änderung
    bra.s SetOK                     ;OK ausgeben

  Gadget14:
    cmp #19,d0                      ;"CLI" angeklickt?
    bne.s Gadget15
    lea NewCli(pc),a5               ;Dann "NewCli"
    jsr MyEx(pc)                    ;starten
    beq.s SetOK                     ;und OK ausgeben

  Gadget15:
    cmp #20,d0                      ;"Info" angeklickt?
    bne.s SetGad
    lea Info(pc),a0                 ;Dann Info-Alert
    jsr Alert(pc)                   ;ausgeben

  ;                             ===> GADGETS SETZEN
  SetOK:
    move.l #OK,Error                ;Error auf OK
  SetGad:
    move ColMask(pc),d6             ;RGB-Abstufung holen
    move RGB(pc),d5                 ;und mit bisheriger
    eor d5,d6                       ;Einstellung der RGB-
    btst #8,d6                      ;Schalter verknüpfen
    beq.s TstGad2
    moveq #111-2,d7                 ;Wenn Flag gesetzt
    jsr Rect(pc)                    ;ist, dann Zustand
    bchg #8,d5                      ;des "R" ändern
  TstGad2:
    btst #4,d6
    beq.s TstGad3
    move #111+24-2,d7               ;Wenn Flag gesetzt
    jsr Rect(pc)                    ;ist, dann Zustand
    bchg #4,d5                      ;des "G" ändern
  TstGad3:
    btst #0,d6
    beq.s TstGad4
    move #111+48-2,d7               ;Wenn Flag gesetzt
    jsr Rect(pc)                    ;ist, dann Zustand
    bchg #0,d5                      ;des "B" ändern
  TstGad4:
    move d5,RGB                     ;RGB-Zustand merken
    move.b OldFlags(pc),d6          ;alter Zustand
    move.b ParFlags(pc),d5          ;neuer Zustand
```

```
	eor.b d5,d6			;Gad-Flags verknüpfen
	btst #0,d6			;Flag geändert?
	beq.s TstGad5			;Nein, verzweigen!
	move.l #$1820180,d0
	lea ShadeBack(pc),a5		;Testen,Hinter-
	btst #0,d5			;grund oder Textfarbe
	beq.s SetGad4			;für Abstufung ange-
	swap d0				;wählt ist, und dem-
	lea ShadeText(pc),a5		;entsprechend Gadget-
SetGad4:				;Text und Farb-
	move d0,ColReg1+4		;register setzen
	swap d0
	move d0,ColReg2
	jsr Print(pc)
TstGad5:
	btst #1,d6			;Flag geändert?
	beq.s TstGad6			;Nein, verzweigen!
	moveq #$67,d0
	lea IntOff1(pc),a5		;Testen,ob Auto- oder
	btst #1,d5			;CLI-Abschaltung an-
	beq.s SetGad5			;gewählt ist, und
	moveq #$60,d0			;dementsprechend
	lea IntOff2(pc),a5		;Gadget-Text setzen
SetGad5:				;und Interrupt-
	move.b d0,IntOnOff		;Routine anpassen
	jsr Print(pc)			;$60=bra; $67=beq
TstGad6:
	tst.b d6			;Flag geändert?
	bpl.s ChkErr			;Nein, verzweigen!
	moveq #$65,d0
	moveq #-1,d1			;Testen, ob Scrolling
	lea ScrollOn(pc),a5		;ein- oder ausge-
	tst.b d5			;schaltet ist, und
	bpl.s SetGad6			;dementsprechend
	moveq #$60,d0			;Gadget-Text setzen
	moveq #$2f,d1			;und Interrupt-
	lea ScrollOff(pc),a5		;Routine anpassen
SetGad6:				;$60=bra; $65=bcs
	move.b d0,ScrollOnOff
	move.b d1,VertPos
	jsr Print(pc)
ChkErr:
	move.l Error(pc),d0		;Fehler?
	beq.s GoMain			;Nein, Hauptschleife
	move.l d0,a4
	jsr PrintStat(pc)		;Fehler ausgeben
GoMain:
	bra Main
```

```

;					===> PROGRAMM BEENDEN
ExitMain:
	jsr Sure(pc)			;Sicherheitsabfrage
	beq Main
	move.l GfxBase(pc),a6
	moveq #1,d7
	move.l ColBf(pc),d6
RestoreCol:				;Alte Farben
	move.l WbViewPort(pc),a0	;wiederherstellen
	move d7,d0
	move d6,d3
	and #$f,d3
	lsr #4,d6
	move d6,d2
	and #$f,d2
	lsr #4,d6
	move d6,d1
	and #$f,d1
	swap d6
	CALL SetRGB4
	dbra d7,RestoreCol

	move.l IntuitionBase(pc),a6
	move.l MyWindow(pc),a0		;Window schließen
	CALL CloseWindow

DeleteFiles:
	jsr DeleteTx(pc)
	move.l #MyEdName,d1
	CALL DeleteFile			;Ed löschen

	move.l ExecBase,a6
	move.l IntuitionBase(pc),a1
	CALL CloseLibrary		;Librarys schließen
	move.l GfxBase(pc),a1
	CALL CloseLibrary
	move.l DosBase(pc),a1
	CALL CloseLibrary

	EXIT_AMIGA


;### UNTERPROGRAMME ###

;					===> PROPS EINSTELLEN
SetProp:
	swap d6				;Farben in d6
SetProp2:
	lea Gadgets,a0
	moveq #5,d0
	moveq #17,d1
	moveq #0,d7
SetPropLoop:
	move d1,4(a0)			;LeftEdge
	move #22,6(a0)			;TopEdge
	move #14,8(a0)			;Width
	move #64,10(a0)			;Height
	move #3,14(a0)			;Activation
	move #3,16(a0)			;GadgetType
	move d7,38(a0)			;GadgetID
	rol.l #4,d6
	move d6,48(a0)
	and #$f000,48(a0)
	or #$0800,48(a0)
	move #$fff,52(a0)		;VertBody
	add #44,a0
	move #5,(a0)			;Flags
```

```
         move.l a0,-10(a0)            ;SpecialInfo
         add #22,a0
         move.l a0,-48(a0)            ;GadgetRender
         addq #8,a0
         move.l a0,-74(a0)            ;NextGadget
         addq #1,d7
         add #24,d1
         cmp #3,d0                    ;3 Gadgets ge-
         bne.s NextProp               ;setzt?
         add #110,d1                  ;dann Pos. ändern
         rol.l #4,d6                  ;Nächste Farbe
        NextProp:
         dbra d0,SetPropLoop          ;Nächstes Gadget
         rts

        ;                       ===> PROPS REFRESHEN
        RefreshProp:
         lea Gadgets,a0               ;Prop-Gadgets
         move.l MyWindow(pc),a1
         sub.l a2,a2
         moveq #6,d0                  ;6 Gadgets refreshen
         move.l IntuitionBase(pc),a6
         jmp _LVORefreshGList(a6)

        ;                       ===> STATUSZEILE LÖSCHEN
        ClrStat:
         lea Clear(pc),a5             ;Spaces
         jmp Print(pc)                ;ausgeben

        ;                       ===> STATUS AUSGEBEN
        PrintStat:
         jsr ClrStat(pc)              ;Statuszeile löschen
         move.l a4,a5                 ;Status ausgeben

        ;                       ===> TEXT AUSGEBEN
        Print:
         movem.l d0-d7/a0-a4/a6,-(sp) ;Register ablegen
         moveq #1,d7                  ;Schrift + Schatten
        PrintLoop:
         move.l a5,a4                 ;Zeiger auf Text
         move.l d7,d0                 ;JAM1 / JAM2
         move.l MyRastPort(pc),a1
         move.l GfxBase(pc),a6
         CALL SetDrMd                 ;Zeichenmodus
         move.l d7,d0                 ;weiß / schwarz
         addq #1,d0
         CALL SetAPen                 ;Zeichenfarbe
         clr.l d0
         clr.l d1                     ;Position setzen
         move.b (a4)+,d0
         move.b (a4)+,d1
         add.b d7,d0                  ;Schrift verschieben
         CALL Move                    ;Position setzen
         moveq #-1,d0
         move.l a4,a0                 ;Anzahl der aus-
        GetTxEnd:                     ;zugebenden Zeichen
         addq #1,d0                   ;ermitteln
         tst.b (a4)+
         bne.s GetTxEnd
         CALL Text                    ;Text ausgeben
         dbra d7,PrintLoop            ;auch Schatten setzen
         move.l a4,a5                 ;neuer Zeiger nach a5
         movem.l (sp)+,d0-d7/a0-a4/a6;Register holen
         rts

        ;                       ===> RECHTECK INVERTIEREN
        Rect:
         moveq #RP_COMPLEMENT,d0      ;invertieren
         move.l MyRastPort(pc),a1
         move.l GfxBase(pc),a6
         CALL SetDrMd                 ;Zeichenmodus
         move.l d7,d0
         moveq #23+60-8,d1
         moveq #11,d2                 ;Positionen setzen
         add.l d7,d2                  ;und
         moveq #23+60-8+10,d3
         jmp _LVORectFill(a6)         ;Rechteck invertieren

        ;                       ===> SICHERHEITSABFRAGE
        Sure:
         move.b SaveFlag(pc),d0       ;Text nicht geändert?
         bne.s ExitSure               ;Dann kein Alert
         move.b TxClrFlag(pc),d0      ;Kein Text?
         bne.s ExitSure               ;Dann kein Alert!
         lea SureTx(pc),a0
         jsr Alert(pc)                ;Alert ausgeben
        ExitSure:
         tst.b d0                     ;Taste prüfen
         rts

        ;                       ===> ALERT DARSTELLEN
        Alert:
         move.l ExecBase,a6
         CALL Disable                 ;Interrupts aus
         moveq #0,d0                  ;Recovery
         moveq #33,d1                 ;Höhe: 33
         move.l IntuitionBase(pc),a6
         CALL DisplayAlert            ;Alert darstellen
         move.l ExecBase,a6
         jmp _LVOEnable(a6)           ;Interrupts ein

        ;                       ===> FILES KOPIEREN
        FileCopy:
         tst.b FileName               ;Filename vorhanden?
         bne.s NoNameErr
         move.l #AskTxName,Error      ;Nein, Fehler
         bra.s NoClose
        NoNameErr:
         jsr OpenOld(pc)              ;Quelle öffnen
         beq.s NoClose
         move.l d7,d1
         move.l d0,d7                 ;Handle nach d7
         jsr OpenNew(pc)              ;Ziel öffnen
         beq.s MyClose
         move.l d0,d6                 ;Handle nach d6
        FileCopyLoop:
         move.l d7,d1                 ;Quell-Handle
         move.l #FileBf,d2            ;Startadresse
         move.l #BfSize,d3            ;max. 20000 Bytes
```

```
    CALL Read
    move.l d0,d3                    ;Länge = neue Länge
    beq.s ExitFileCopy2             ;Ende?
    not.l d0
    beq.s ExitFileCopy              ;Fehler?
    move.l d6,d1                    ;Ziel-Handle
    move.l #FileBf,d2               ;Startadresse
    CALL Write
    not.l d0
    bne.s FileCopyLoop              ;Fehler? Nein, weiter
ExitFileCopy:
    jsr TestDosErr(pc)              ;Fehler prüfen
ExitFileCopy2:
    move.l d6,d1
    CALL Close                      ;Ziel schließen
    bra.s MyClose                   ;Quelle schließen

;                           ===> FILE EINLESEN
CopyFileToMem:
    jsr OpenOld(pc)                 ;Öffnen
    beq.s NoClose                   ;Fehler?
    move.l a5,d2
    move.l d6,d3                    ;Parameter setzen
    move.l d0,d7
    move.l d0,d1
    CALL Read
    bra.s ExitCopy                  ;Schließen

;                           ===> FILE SCHREIBEN
CopyMemToFile:
    jsr OpenNew(pc)                 ;Öffnen
    beq.s NoClose                   ;Fehler?
    move.l a5,d2
    move.l d6,d3                    ;Parameter setzen
    move.l d0,d7
    move.l d0,d1
    CALL Write
ExitCopy:
    move.l d0,d6                    ;Länge nach d6
    not.l d0
    jsr TestDosErr(pc)              ;Fehler?

;                           ===> CLOSE
MyClose:
    move.l d7,d1                    ;Filehandle aus d7
    CALL Close                      ;Schließen
NoClose:
    clr.l d0                        ;für Abfrage löschen
    tst.l Error                     ;Fehler prüfen
    rts

;                           ===> OPEN
OpenNew:
    move.l #MODE_NEWFILE,d2         ;Neues File öffnen
    bra.s OpenNewJmp
OpenOld:
    move.l #MODE_OLDFILE,d2         ;Altes File öffnen
OpenNewJmp:
    move.l DosBase(pc),a6
    CALL Open
TestDosErr:
    tst.l d0                        ;Fehler?
    bne.s NoDosErr                  ;Nein, verzweigen
SetDosErr:
    move.l #DosErr,Error            ;Ja, "DOS-Fehler"
NoDosErr:
    tst.l d0                        ;Fehler prüfen
    rts

;                           ===> TEXT VORHANDEN?
TestTx:
    tst.b TxClrFlag
    beq.s NoClose
    move.l #NoTx,Error              ;Nein, dann Error
    bra.s NoClose

;                           ===> TEXT LÖSCHEN
DeleteTx:
    move.l #BootTxName,d1
    move.l DosBase(pc),a6
    jmp _LVODeleteFile(a6)          ;File löschen

;                           ===> EXECUTE
MyEx:
    move.l #NIL,d1
    jsr OpenOld(pc)                 ;Filehandle erzeugen
    beq.s NoClose                   ;Fehler?
    move.l d0,d7                    ;Handle retten
    move.l a5,d1
    clr.l d2                        ;Parameter setzen
    move.l d0,d3
    CALL Execute
    jsr TestDosErr(pc)
    bra.s MyClose                   ;Schließen

;                           ===> PARAMETERFILE
SetParName:
    lea FileName,a4
    move.l a4,d1
ParNameLoop:
    tst.b (a4)+                     ;"&" an den aktuellen
    bne.s ParNameLoop               ;Filenamen anhängen
    clr.b (a4)
    move.b #"&",-(a4)
    lea ActualCol(pc),a5            ;Startadresse
    moveq #OldFlags-ActualCol,d6;Länge des Files
    rts


;### BOOTPROGRAMM ###

BootBlock:
    DC.B "DOS",0
ChkSum:
    DC.L 0,$0370

BootPrg:
    lea IntPrg(pc),a0
    lea $100,a1                     ;Interrupt-Routine
    move #BootTx-IntPrg-1,d0        ;nach $100
CopyIntPrg:                         ;kopieren
    move.b (a0)+,(a1)+
    dbra d0,CopyIntPrg

    lea GfxName(pc),a1              ;Library öffnen
    CALL OldOpenLibrary
    move.l d0,a6                    ;alten Copper-
```

```
        move.l 38(a6),GfxName-IntPrg+$100 ;Zeiger retten
        lea TextAttr-IntPrg+$100,a0
        CALL OpenFont                   ;Font öffnen
        move.l d0,a0                    ;Zeiger auf Font-
        move.l 34(a0),d6                ;Daten nach d6
        move.l ExecBase,a6
        move.l #ScreenSize,d0           ;Speicher
        move.l #MEMF_CHIP|MEMF_CLEAR,d1 ;für Screen
        CALL AllocMem
                                        ;reservieren
        move.l d0,GfxName-IntPrg+$104   ;Zeiger ablegen
        move.l d0,a0
        move d0,ScreenPtLo-IntPrg+$100  ;Zeiger in
        swap d0                         ;Copper-Liste
        move d0,ScreenPtHi-IntPrg+$100  ;einbauen

        lea BootTx(pc),a1               ;Textanfang
        moveq #14,d2                    ;15 Zeilen
LineOut:
        moveq #39,d1                    ;40 Zeichen/Zeile
CharOut:
        clr.l d0                        ;d0 löschen
        move.b (a1)+,d0                 ;Zeichen in d0
        add.l d6,d0                     ;+ Zeiger auf Font
        move.l d0,a2                    ;d0 nach a2
        moveq #7,d0
SetChar:
        move.b (a2),(a0)                ;Zeichendaten auf
        move.b (a2),40(a0)              ;den Screen bringen
        add #80,a0                      ;nächstes Screenbyte
        add #$c0,a2                     ;nächstes Datenbyte
        dbra d0,SetChar
        sub #639,a0
        dbra d1,CharOut                 ;nächstes Zeichen
        add #600,a0
        dbra d2,LineOut                 ;nächste Zeile

        lea BootTx-IntPrg+$100,a0       ;Ende der CopperListe
        moveq #15,d0            ;ab Rasterzeile 15
ColValue1:
        move #$000,d1           ;Anfangsfarbwert
        moveq #15,d2            ;16 Abstufungen
MakeCol:
        move.b d0,(a0)+         ;Rasterzeile ablegen
        move.b #9,(a0)+         ;x-Rasterpositon
ColReg1:
        move.l #$fffe0180,(a0)+         ;Farbregister
        move d1,(a0)+           ;Farbwert ablegen
        add.b #16,d0
ColSub:
        sub #0,d1               ;nächste Abstufung
        dbra d2,MakeCol
        move.l #$fffffffe,(a0)  ;CopList-Ende: warten

        move.l $6c,Font-IntPrg+$100 ;Int-Vektor retten
        move.l #$100,$6c                ;und neu setzen
        move.l #CopList-IntPrg+$100,$dff080
                                        ;neue CopList
        lea DosName(pc),a1
        CALL FindResident               ;DOS-Library
        move.l d0,a0                    ;initialisieren
        move.l 22(a0),a0
        moveq #0,d0
        rts

DosName:
        DC.B "dos.library",0

;                               ===> INTERRUPT-ROUTINE
IntPrg:
        movem.l d0-d7/a0-a6,-(a7)       ;Register ablegen
        btst #5,$dff01f                 ;Vertical-Blanking?
        beq.s ExitInt
        cmp.b #$30,VertPos-IntPrg+$100  ;Text noch nicht
ScrollOnOff:                            ;oberste Zeile
        bcs.s NoScroll                  ;erreicht?
        subq.b #1,VertPos-IntPrg+$100   ;Dann scrollen!
NoScroll:
        move.l ExecBase,a6
        lea ConTask(pc),a1              ;Task für CLI-Fenster
        CALL FindTask                   ;vorhanden?
        tst.l d0
IntOnOff:
        beq.s ExitInt
        move.l Font(pc),$6c             ;alte Int-Routine
        move.l GfxName(pc),$dff080      ;alte Cop-List
        move #%1000000000100000,$dff096 ;Sprites ein
        move.l #ScreenSize,d0
        move.l GfxName+4(pc),a1 ;Speicher für
        CALL FreeMem            ;Screen abmelden
ExitInt:
        movem.l (a7)+,d0-d7/a0-a6       ;Register holen
        DC.W $4ef9              ;Code für "jmp";
                                ;springt zur alten
Font:                           ;Interrupt-Routine
        DC.B "topaz.font"       ;mit Adr. in Font
TextAttr:
        DC.L Font-IntPrg+$100           ;ta_Name
        DC.W 8                          ;ta_YSize
        DC.B 0                          ;ta_Style
        DC.B 0                          ;ta_Flags
ConTask:
        DC.B "CON",0
GfxName:
        DC.B "graphics.library"

CopList:
        DC.B 0
        DC.B $096                       ;dmacon
        DC.W %1000000110000000
        DC.W $096                       ;dmacon
        DC.W %0000000000100000
ColReg2:
        DC.W $182                       ;color0x
ColValue2:
        DC.W $0000
        DC.W $08e                       ;diwstrt
VertPos:
        DC.W $ff81
        DC.W $090                       ;diwstop
        DC.W $3fc1
        DC.W $092                       ;ddfstrt
        DC.W $0038
```

```
    DC.W $094                           ;ddfstop
    DC.W $00d0
    DC.W $108                           ;bpl1mod
    DC.W $0000
    DC.W $102                           ;bplcon1
    DC.W $0000
    DC.W $104                           ;bplcon2
    DC.W %100100
    DC.W $0e0                           ;bplpt[0]
ScreenPtHi:
    DC.W $0000
    DC.W $0e2
ScreenPtLo:
    DC.W $0000
    DC.W $100                           ;bplcon0
    DC.W %0001001000000000
BootTx:


;### INITALISIERUNG ###

Startup:
    INIT_AMIGA
;                               ===> LIBRARYÖFFNEN

    lea DosName(pc),a1                  ;Dos-Library
    CALL OldOpenLibrary                 ;öffnen
    move.l d0,DosBase
    lea GfxName(pc),a1                  ;Graphics-
                                         Library
    CALL OldOpenLibrary                 ;öffnen
    move.l d0,GfxBase
    lea IntuitionName(pc),a1            ;Intuition-
                                         Library
    CALL OldOpenLibrary                 ;öffnen
    move.l d0,IntuitionBase
;                               ===> ED LADEN UND
                                     ÄNDERN

    clr.l Error                         ;Fehler-Var.
                                         löschen
    move.l #EdName,d1
    lea FileBf,a5
    move.l #BfSize,d6                   ;Ed laden
    jsr CopyFileToMem(pc)
    bne.s DoEdAlert               ;Fehler? Dann Alert!
ModifyEd:
    moveq #2,d0                   ;Angaben für
    lea MyEdRAW(pc),a0            ;RAW-Fenster suchen
GotOldRAW:
    move.b (a5)+,d1
    cmp.l #FileBf+BfSize,a5             ;Kein "RAW"
                                         gefunden?
    beq.s DoEdAlert                     ;Dann Alert!
    bclr #5,d1                          ;Groß-/
                                         Kleinschr.aus
    cmp.b (a0)+,d1
    bne.s ModifyEd
    dbra d0,GotOldRAW             ;Neue RAW- Angaben
    moveq #EdName-MyEdRaw-4,d0    ;kopieren
CopyNewRAW:
    move.b (a0)+,(a5)+
    dbra d0,CopyNewRAW
    move.l #MyEdName,d1                 ;Ed ins RAM
    lea FileBf,a5
    jsr CopyMemToFile(pc)
    beq.s MakeGad
DoEdAlert:
    lea EdAlert(pc),a0            ;Ed mach Probleme
    jsr Alert(pc)                 ;Alert ausgeben
    bra DeleteFiles               ;und aussteigen

;                               ===> GADGET-STRUCT
                                     SETZEN
```

```
MakeGad:
    move.l IntuitionBase(pc),a6
    move.l 56(a6),a5
    lea 44(a5),a5                   ;Zeiger auf
    move.l a5,WbViewPort            ;View-Port holen
    move.l 4(a5),a5
    move.l 4(a5),a5                 ;Workbench-Farben
    move.l (a5),d6                  ;retten
    move.l d6,ColBf                 ;Prop-Einstellung
    jsr SetProp(pc)                 ;Props setzen
    moveq #15,d0                    ;16 Bool-Gadgets
    lea GadgetData(pc),a1           ;Zeiger auf Daten
SetBool:
    move.b (a1)+,5(a0)                    ;LeftEdge
    move.b (a1)+,7(a0)                    ;TopEdge
    clr.l d1
    move.b (a1)+,d1
    move d1,8(a0)                         ;Width
    subq #1,d1
    move d1,64(a0)                        ;xy
    move d1,68(a0)                        ;xy
    move #11,10(a0)                       ;Height
    move #10,70(a0)                       ;xy
    move #10,74(a0)                       ;xy
    addq #3,14(a0)                        ;Activation
    cmp #10,d0
    bcs.s NoInv
    addq #3,12(a0)                  ;Invertierung aus
NoInv:
    addq #1,16(a0)                        ;GadgetType
    move d7,38(a0)                        ;GadgetID
    move.l #$1000005,48(a0)               ;Front/Back,
                                           Count
    add #44,a0
    move.l a0,-26(a0)                     ;GadgetRender
    add #16,a0
    move.l a0,-8(a0)                      ;Border-Data
    add #20,a0
    move.l a0,-80(a0)                     ;Next Gadget
    addq #1,d7
    dbra d0,SetBool                       ;nächstes
                                           Gadget
    clr.l -80(a0)
    addq #2,-74(a0)                       ;Position des
    subq #2,-34(a0)                       ;String-Gadgets
    addq #2,-76(a0)                       ;korrigieren
    subq #2,-36(a0)
    addq #3,-64(a0)                       ;Typ = String
    move.l a0,-46(a0)
    move #41,10(a0)                       ;MaxChars
    add #36,a0
    move.l a0,-36(a0)                     ;Buffer

;                               ===> WINDOW ÖFFNEN
    lea NewWindow(pc),a0
    CALL OpenWindow                       ;Fenster öffnen
    move.l d0,MyWindow              ;und Zeiger retten
    beq DeleteFiles                 ;Nicht zu öffnen?
    move.l d0,a0
    move.l 50(a0),MyRastPort        ;RastPort holen
    lea GadgetTx(pc),a5
    moveq #30,d0
PrintGadTx:
    jsr Print(pc)                   ;Gadget-Beschriftung
    dbra d0,PrintGadTx              ;ausgeben
    bra Main


;### DATEN ###

IntuitionName:
    DC.B "intuition.library",0

NewWindow:
```

```
 DC.W 180,0         ;LeftEdge, TopEdge
 DC.W 280,200       ;Width, Height
 DC.B 0,1           ;DetailPen, BlockPen
 DC.L $80748        ;IDCMPFlags
 DC.L $1100e        ;Flags
 DC.L Gadgets       ;FirstGadget
 DC.L 0             ;Checkmark
 DC.L WindowName    ;Title
 DC.L 0             ;Screen
 DC.L 0             ;BitMap
 DC.W 100,100       ;MinWidth, MinHeight
 DC.W 100,100       ;MaxWidth, MaxHeight
 DC.W 1             ;Type
GadgetData:
 DC.B 111-2,83-8,8+4     ;R
 DC.B 111+24-2,83-8,8+4 ;G
 DC.B 111+48-2,83-8,8+4 ;B
 DC.B 95-2,47-8,11*8+4  ;Abstufung
 DC.B 172-2,105-8,5*8+4 ;Abschaltung
 DC.B 172-2,117-8,5*8+4 ;Scrolling
 DC.B 127-2,19-8,3*8+4  ;<->
 DC.B 16-2,156-8,124    ;editieren
 DC.B 144-2,156-8,124   ;löschen
 DC.B 16-2,132-8,31*8+4 ;ansehen
 DC.B 16-2,144-8,31*8+4 ;Boot-Block
 DC.B 16-2,168-8,124    ;laden
 DC.B 144-2,168-8,124   ;speichern
 DC.B 232-2,183-8,4*8+4 ;CLI
 DC.B 232-2,195-8,4*8+4 ;Info
 DC.B 96-2,183-8,16*8+4 ;Filename
GadgetTx:
 DC.B 4,19,"Hintergrund",0
 DC.B 127,19,"<->",0
 DC.B 186,19," Textfarbe ",0
 DC.B 20,93,"R",0
 DC.B 20+24,93,"G",0
 DC.B 20+48,93,"B",0
 DC.B 203,93,"R",0
 DC.B 203+24,93,"G",0
 DC.B 203+48,93,"B",0
 DC.B 87,36,"Abstufung auf",0
 DC.B 95,64,"RGB-Auswahl",0
 DC.B 87,72,"für Abstufung",0
 DC.B 111,83,"R",0
 DC.B 111+24,83,"G",0
 DC.B 111+48,83,"B",0
 DC.B 66,105,"Abschaltung:",0
 DC.B 66,117,"Scrolling:",0
 DC.B 140-68,132,"Boot-Text ansehen",0
 DC.B 140-116,144,"Boot-Block auf df0:
                   schreiben",0
 DC.B 76-56,156,"Text edieren",0
 DC.B 204-48,156,"Text löschen",0
 DC.B 76-40,168,"Text laden",0
 DC.B 204-56,168,"Text speichern",0
 DC.B 16,183,"Textname:",0
 DC.B 236,183,"CLI",0
 DC.B 232,195,"Info",0
 DC.B 32,195,"Status:",0
OK:         DC.B 96,195,"OK!",0
ShadeBack:  DC.B 95,47,"Hintergrund",0
IntOff1:    DC.B 176,105,"auto",0
ScrollOn:   DC.B 180,117,"ein",0
ShadeText:  DC.B 95,47," Textfarbe ",0
IntOff2:    DC.B 172,105," CLI ",0
ScrollOff:  DC.B 180,117,"aus",0
Clear:      DC.B 95,195,"                  ",0
DosErr:     DC.B 96,195,"DOS-Fehler!",0
AskTxName:  DC.B 96,195,"Textname?",0
NoTx:       DC.B 96,195,"Kein Text!",0
CutTx:       DC.B 96,195,"Schnipp!",0
BootStat1:  DC.B 96,195,"BootBlock...",0
BootStat2:  DC.B 96,195,"c/BootOff...",0
DiskErr:    DC.B 96,195,"Disk-Fehler!",0
WriteProt:  DC.B 96,195,"Schreibschutz?",0
WindowName:   DC.B "BootLetter 1.0    (c)K",0
TD_NAME:      DC.B "trackdisk.device",0
NIL:          DC.B "NIL:",0
NewCli:       DC.B "newcli con:0/0/640/159/
                    CLI",0
MyEdRAW:      DC.B "RAW:145/0/350/141/Ed",0
EdName:       DC.B "SYS:c/ed",0
BootOffName:  DC.B "c/BootOff",0
MyEdName:     DC.B "ram:ed",0
RunEd:        DC.B "ram:ed ram:boot.tx",0
EdAlert:
 DC.B 0,248,18,"Ed macht Probleme! ",0,0
SureTx:
 DC.B 0,252,14,"Text gespeichert?",0,1
 DC.B 0,16,22,"linke Maustaste: JA",0,1
 DC.B 1,168,22,"rechte Maustaste:
                 ABBRUCH",0,0
Info:
 DC.B 0,108,18,"(c) KICKSTART          "
 DC.B "Industriestr. 28, Eschborn      ",0,0
BootOff:
 DC.B $00,$00,$03,$F3,$00,$00,$00,$00
 DC.B $00,$00,$00,$01,$00,$00,$00,$00
 DC.B $00,$00,$00,$00,$00,$00,$00,$05
 DC.B $00,$00,$03,$E9,$00,$00,$00,$05
 DC.B $31,$FC,$4E,$71,$01,$28,$0C,$B8
 DC.B $00,$00,$01,$00,$00,$6C,$67,$F6
 DC.B $4E,$75,$00,$00,$00,$00,$03,$F2
 DC.B $00,$00,$03,$F2
 ALIGN
;### VARIABLEN UND FLAGS ###
ActualCol:      DC.L 0
ColMask:        DC.W 0
ParFlags:       DC.B 0
OldFlags:       DC.B 0
CharBf:         DC.B 0
TxClrFlag:      DC.B -1
RGB:            DC.W 0
ColBf:          DC.L 0
WbViewPort:     DC.L 0
MyWindow:       DC.L 0
MyRastPort:     DC.L 0
DosBase:        DC.L 0
GfxBase:        DC.L 0
IntuitionBase:  DC.L 0
Error:          DC.L 0
SaveFlag:       DC.B 0
;### RESERVIERTER SPEICHER ###
 BSS
ReplyPort:  DS.B 34
IOExtTD:    DS.B 56
Gadgets:    DS.B GadgetSize
FileBf:     DS.B BfSize
 END
```

Listing 1: BOOTLETTER, der universelle Vorspanngenerator aus dem Bootblock

```
'BootLetter.BAS  (c) KICKSTART 1988

nam$="BootLetter":OPEN "o",#1,nam$:z=16
WHILE q$<>"#":READ hx$:z=z+1:s=z:x=1:y=0
GetHx:
 b=VAL("&h"+MID$(hx$,x,2)):x=x+1:q$=" "
 WHILE q$=" ":x=x+1:q$=MID$(hx$,x,1):WEND
 y=y+1:IF y>10 OR q$="#" THEN Check
 s=s+b*y AND 255:PRINT#1,CHR$(b);:GOTO GetHx
Check:
 IF s<>b THEN PRINT"Fehler:",z,"DATA ";hx$
WEND:CLOSE:KILL nam$+".info"

'Zeilennummern können weggelassen werden,
'vereinfachen aber die Fehlersuche!

17    DATA 00 00 03 F3 00 00 00 00 00 00  E6
18    DATA 00 02 00 00 00 00 00 00 00 01  20
19    DATA 40 00 03 B6 40 00 15 63 00 00  1F
20    DATA 03 E9 00 00 03 B6 4E FA 09 4C  77
21    DATA 4B F9 00 00 00 62 7E 01 E1 86  3D
22    DATA 20 7A 0E AA 30 07 0A 40 00 01  66
23    DATA 12 2D 00 30 E8 09 8C 01 E9 86  4A
24    DATA 14 2D 00 7A E8 0A 8C 02 E9 86  83
25    DATA 16 2D 00 C4 E8 0B 8C 03 2C 7A  9F
26    DATA 0E 94 4E AE FE E0 DA FC 00 DE  AA
27    DATA 51 CF FF CA 23 C6 00 00 0E AA  A4
28    DATA 08 3A 00 00 0E 62 66 02 48 46  48
29    DATA 30 3A 0E 58 8C 40 33 C6 00 00  1C
30    DATA 08 46 48 46 33 C6 00 00 09 24  FE
31    DATA C0 7C 01 11 33 C0 00 00 08 5E  91
32    DATA 20 7A 0E 4A 20 68 00 56 2C 78  82
33    DATA 00 04 4E AE FE 8C 4A 80 67 00  AE
34    DATA FF 7E 22 40 2A 40 4E AE FE 86  91
35    DATA 30 2D 00 16 0C B8 00 00 01 00  9A
36    DATA 00 6C 66 08 31 FC 4E 71 01 28  6E
37    DATA 42 80 B0 7C 02 00 67 00 04 22  BA
38    DATA B0 7C 00 40 66 00 FF 50 2A 6D  01
39    DATA 00 1C 30 2D 00 26 B0 7C 00 0F  CD
40    DATA 64 06 42 39 00 00 0E D6 42 B9  E0
41    DATA 00 00 0E D2 13 FA 0D DE 00 00  21
42    DATA 0E B1 B0 7C 00 06 66 08 0A 79  DC
43    DATA 0F 00 00 00 0E AE B0 7C 00 07  8A
44    DATA 66 08 0A 79 00 F0 00 00 0E AE  8E
45    DATA B0 7C 00 08 66 08 0A 79 00 0F  C7
46    DATA 00 00 0E AE B0 7C 00 09 66 08  96
47    DATA 08 79 00 00 00 00 0E B0 B0 7C  13
48    DATA 00 0A 66 08 08 79 00 01 00 00  9C
49    DATA 0E B0 B0 7C 00 0B 66 08 08 79  ED
50    DATA 00 07 00 00 0E B0 B0 7C 00 0C  CE
51    DATA 66 0E 2C 3A 0D 76 4E BA 04 20  7C
52    DATA 4E BA 04 98 42 80 B0 7C 00 0F  F2
53    DATA 66 16 4E BA 06 08 66 10 4B FA  98
54    DATA 07 7A 1E 15 1A BC 00 60 4E BA  CB
55    DATA 06 5E 1A 87 B0 7C 00 10 66 00  D1
56    DATA 01 38 4E BA 05 EA 66 00 01 30  C3
57    DATA 49 FA 0B 9B 4E BA 04 80 42 B9  8D
58    DATA 00 00 07 AE 42 80 32 3C 00 FF  85
59    DATA 41 FA 06 28 D0 98 64 02 52 80  70
60    DATA 51 C9 FF F8 72 FF 92 80 23 C1  F3
61    DATA 00 00 07 AE 43 F9 00 00 00 08  7F
62    DATA 23 79 00 00 00 04 00 10 4E AE  75
63    DATA FE 9E 43 F9 00 00 00 2A 70 00  66
64    DATA 42 81 41 FA 0B A7 4E AE FE 44  78
65    DATA 4A 80 66 00 00 A6 43 F9 00 00  3E
66    DATA 00 2A 23 7C 00 00 00 08 00 0E  BB
67    DATA 33 7C 00 0F 00 1C 42 A9 00 24  D0
68    DATA 4E AE FE 38 43 F9 00 00 00 2A  91
69    DATA 7C 1C 4A A9 00 20 66 42 33 7C  B8
70    DATA 00 03 00 1C 23 7C 00 00 07 AA  36
71    DATA 00 28 23 7C 00 00 04 00 00 24  74
72    DATA 42 A9 00 2C 4E AE FE 38 43 F9  ED
73    DATA 00 00 00 2A 1C 29 00 1F 66 18  F1
74    DATA 33 7C 00 04 00 1C 42 A9 00 24  AB
75    DATA 4E AE FE 38 43 F9 00 00 00 2A  98
76    DATA 1C 29 00 1F 33 7C 00 09 00 1C  7D
77    DATA 42 A9 00 24 4E AE FE 38 43 F9  D2
78    DATA 00 00 00 08 4E AE FE 98 43 F9  CF
79    DATA 00 00 00 2A 4E AE FE 3E 4A 06  49
80    DATA 67 20 23 FC 00 00 0D 36 00 00  5B
81    DATA 0E D2 BC 3C 00 1C 67 00 01 94  71
82    DATA 23 FC 00 00 0D 27 00 00 0E D2  4A
83    DATA 60 00 01 86 08 3A 00 01 0C 38  F6
84    DATA 67 00 01 72 49 FA 0A 98 4E BA  D7
85    DATA 03 6E 22 3C 00 00 0D AC 4B FA  AC
86    DATA 0B E0 7C 3C 4E BA 04 6A 67 00  72
87    DATA 01 56 B0 7C 00 0D 66 0E 4E BA  8E
88    DATA 03 48 4B FA 0B 17 4E BA 04 C8  5B
89    DATA 67 56 B0 7C 00 0E 66 14 4E BA  2C
90    DATA 03 A4 67 0E 4E BA 04 A8 50 F9  DA
91    DATA 00 00 0E B3 60 00 01 28 B0 7C  80
92    DATA 00 11 66 00 00 C6 4E BA 03 18  51
93    DATA 22 3C 00 00 07 42 2E 3C 00 00  C8
94    DATA 0D C4 4E BA 03 AC 66 00 00 AE  72
95    DATA 4E BA 04 A6 4E BA 03 FC 42 14  B6
96    DATA 2C 3A 0B B6 4E BA 02 5E 4E BA  DB
97    DATA 02 D8 50 F9 00 00 0E D6 22 3C  83
98    DATA 00 00 0D C4 4E BA 04 24 42 B9  43
99    DATA 00 00 0E D2 4A 80 67 7C 42 39  84
100   DATA 00 00 0E B3 2E 00 4B FA 06 2C  0B
101   DATA 30 3C 02 57 42 35 00 00 51 C8  A0
102   DATA FF FA 7C 00 22 07 24 3C 00 00  7D
103   DATA 0E B2 76 01 4E AE FF D6 4A 80  1C
104   DATA 67 40 46 80 67 3C 12 3A 0B 68  4D
105   DATA B2 7C 00 0A 67 26 52 06 6B 22  A7
106   DATA BC 7C 00 29 65 0C 23 FC 00 00  D8
107   DATA 0C FE 00 00 0E D2 60 CA 92 3C  0F
108   DATA 00 20 6A 04 92 3C 00 20 1B 81  39
109   DATA 60 FF 60 BA DA FC 00 28 BB FC  A8
110   DATA 00 00 0B A6 66 AC 23 FC A9 62  C7
111   DATA 76 72 00 00 0B A6 4E BA 03 88  41
112   DATA 67 5A B0 7C 00 12 66 36 4E BA  73
113   DATA 03 B4 66 30 4E BA 02 48 22 3C  88
114   DATA 00 00 0D C4 2E 3C 00 00 07 42  CA
115   DATA 4E BA 02 DC 66 1A 4E BA 03 D8  C2
116   DATA 4E BA 03 42 42 14 4A B9 00 00  D7
117   DATA 0E D2 66 08 50 F9 00 00 0E D6  B9
118   DATA 60 1E B0 7C 00 13 66 0A 4B FA  05
119   DATA 09 9B 4E BA 03 96 67 0E B0 7C  64
120   DATA 00 14 66 12 41 FA 0A 4C 4E BA  E3
121   DATA 02 86 23 FC 00 00 0C 89 00 00  7C
122   DATA 0E D2 3C 3A 0A B4 3A 3A 0A B6  0E
123   DATA BB 46 08 06 00 08 67 0A 7E 6D  F3
124   DATA 4E BA 02 32 08 45 00 08 08 06  96
125   DATA 00 04 67 0C 3E 3C 00 85 4E BA  B2
126   DATA 02 20 08 45 00 04 08 06 00 00  6C
127   DATA 67 0C 3E 3C 00 9D 4E BA 02 0E  E6
128   DATA 08 45 00 00 33 C5 00 00 0E B4  35
129   DATA 1C 3A 0A 73 1A 3A 0A 6E BB 06  5E
130   DATA 08 06 00 00 67 28 20 3C 01 82  66
131   DATA 01 80 4B FA 08 3B 08 05 00 00  37
132   DATA 67 06 48 40 4B FA 08 4A 33 C0  F5
133   DATA 00 00 08 54 48 40 33 C0 00 00  3A
134   DATA 09 22 4E BA 01 86 08 06 00 01  40
135   DATA 67 1C 70 67 4B FA 08 1F 08 05  0F
136   DATA 00 01 67 06 70 60 4B FA 08 2E  38
137   DATA 13 C0 00 00 08 C8 4E BA 01 64  D7
138   DATA 4A 06 6A 24 70 65 72 FF 4B FA  B9
139   DATA 08 04 4A 05 6A 08 70 60 72 2F  B7
140   DATA 4B FA 08 14 13 C0 00 00 08 B4  62
141   DATA 13 C1 00 00 09 28 4E BA 01 3C  92
142   DATA 20 3A 0A 12 67 06 28 40 4E BA  C9
143   DATA 01 2A 60 00 FB 38 4E BA 01 8A  9A
144   DATA 67 00 FB 30 2C 7A 09 F2 7E 01  A7
145   DATA 2C 3A 09 D8 20 7A 09 D8 30 07  1D
146   DATA 36 06 C6 7C 00 0F E8 4E 34 06  48
147   DATA C4 7C 00 0F E8 4E 32 06 C2 7C  1F
148   DATA 00 0F 48 46 4E AE FE E0 51 CF  1D
```

```
149   DATA FF DC 2C 7A 09 C6 20 7A 09 B2  7E
150   DATA 4E AE FF B8 4E BA 02 50 22 3C  17
151   DATA 00 00 0D B6 4E AE FF B8 2C 78  25
152   DATA 00 04 22 7A 09 A8 4E AE FE 62  5F
153   DATA 22 7A 09 9C 4E AE FE 62 22 7A  CC
154   DATA 09 90 4E AE FE 62 4A B9 00 00  75
155   DATA 00 00 67 0E 4E AE FF 7C 22 79  67
156   DATA 00 00 00 00 4E AE FE 86 70 00  48
157   DATA 4E 75 48 46 41 F9 00 00 00 62  B4
158   DATA 70 05 72 11 7E 00 31 41 00 04  AF
159   DATA 31 7C 00 16 00 06 31 7C 00 0E  07
160   DATA 00 08 31 7C 00 40 00 0A 31 7C  94
161   DATA 00 03 00 0E 31 7C 00 03 00 10  74
162   DATA 31 47 00 26 E9 9E 31 46 00 30  A1
163   DATA 02 68 F0 00 00 30 00 68 08 00  ED
164   DATA 00 30 31 7C 0F FF 00 34 D0 FC  94
165   DATA 00 2C 30 BC 00 05 21 48 FF F6  55
166   DATA D0 FC 00 16 21 48 FF D0 50 48  34
167   DATA 21 48 FF B6 52 47 D2 7C 00 18  FF
168   DATA B0 7C 00 03 66 06 D2 7C 00 6E  68
169   DATA E9 9E 51 C8 FF 94 4E 75 41 F9  21
170   DATA 00 00 00 62 22 7A 08 E2 95 CA  21
171   DATA 70 06 2C 7A 08 EA 4E EE FE 50  D7
172   DATA 4B FA 06 DA 4E FA 00 08 4E BA  09
173   DATA FF F6 2A 4C 48 E7 FF FA 7E 01  59
174   DATA 28 4D 20 07 22 7A 08 BE 2C 7A  EA
175   DATA 08 C2 4E AE FE 9E 20 07 52 40  01
176   DATA 4E AE FE AA 42 80 42 81 10 1C  C4
177   DATA 12 1C D0 07 4E AE FF 10 70 FF  80
178   DATA 20 4C 52 40 4A 1C 66 FA 4E AE  9E
179   DATA FF C4 51 CF FF CA 2A 4C 4C DF  08
180   DATA 5F FF 4E 75 70 02 22 7A 08 80  11
181   DATA 2C 7A 08 84 4E AE FE 9E 20 07  DF
182   DATA 72 4B 74 0B D4 87 76 55 4E EE  80
183   DATA FE CE 10 3A 08 7A 66 0E 10 3A  7B
184   DATA 08 51 66 08 41 FA 07 80 4E BA  08
185   DATA 00 06 4A 00 4E 75 2C 78 00 04  03
186   DATA 4E AE FF 88 70 00 72 21 2C 7A  27
187   DATA 08 4E 4E AE FF A6 2C 78 00 04  FC
188   DATA 4E EE FF 82 4A 39 00 00 07 42  86
189   DATA 66 0C 23 FC 00 00 0C E5 00 00  10
190   DATA 0E D2 60 7C 4E BA 00 8C 67 76  FD
191   DATA 22 07 2E 00 4E BA 00 7A 67 66  C6
192   DATA 2C 00 22 07 24 3C 00 00 07 6C  01
193   DATA 26 3C 00 00 4E 20 4E AE FF D6  8A
194   DATA 26 00 67 18 46 80 67 10 22 06  9A
195   DATA 24 3C 00 00 07 6C 4E AE FF D0  B3
196   DATA 46 80 66 D6 4E BA 00 5E 22 06  D4
197   DATA 4E AE FF DC 60 2E 4E BA 00 44  6A
198   DATA 67 2E 24 0D 26 06 2E 00 22 00  7F
199   DATA 4E AE FF D6 60 12 4E BA 00 28  94
200   DATA 67 1A 24 0D 26 06 2E 00 22 00  59
201   DATA 4E AE FF D0 2C 00 46 80 4E BA  78
202   DATA 00 28 22 07 4E AE FF DC 42 80  61
203   DATA 4A B9 00 00 0E D2 4E 75 24 3C  1F
204   DATA 00 00 03 EE 60 06 24 3C 00 00  6D
205   DATA 03 ED 2C 7A 07 8E 4E AE FF E2  EA
206   DATA 4A 80 66 0A 23 FC 00 00 0C D7  DB
207   DATA 00 00 0E D2 4A 80 4E 75 4A 39  51
208   DATA 00 00 0E B3 67 C6 23 FC 00 00  42
209   DATA 0C F1 00 00 0E D2 60 BA 22 3C  EB
210   DATA 00 00 0D C4 2C 7A 07 5A 4E EE  CC
211   DATA FF B8 22 3C 00 00 0D 6E 4E BA  65
212   DATA FF B6 67 A0 2E 00 22 0D 42 82  96
213   DATA 26 00 4E AE FF 22 4E BA FF B2  41
214   DATA 60 88 49 F9 00 00 07 42 22 0C  F0
215   DATA 4A 1C 66 FC 42 14 19 3C 00 26  48
216   DATA 4B FA 07 06 7C 07 4E 75 44 4F  1E
217   DATA 53 00 00 00 00 00 00 00 03 70  A7
218   DATA 41 FA 00 E8 43 F8 01 00 30 3C  DD
219   DATA 00 AD 12 D8 51 C8 FF FC 43 FA  08
220   DATA 01 40 4E AE FE 68 2C 40 21 EE  0E
221   DATA 00 26 01 6A 41 F8 01 5E 4E AE  6A
222   DATA FF B8 20 40 2C 28 00 22 2C 78  C5
223   DATA 00 04 20 3C 00 00 2A 80 22 3C  E7
224   DATA 00 01 00 02 4E AE FF 3A 21 C0  F6
225   DATA 01 6E 20 40 31 C0 01 A8 48 40  E2
226   DATA 31 C0 01 A4 43 FA 01 42 74 0E  08
227   DATA 72 27 42 80 10 19 D0 86 24 40  F3
228   DATA 70 07 10 92 11 52 00 28 D0 FC  83
229   DATA 00 50 D4 FC 00 C0 51 C8 FF F0  3F
230   DATA 90 FC 02 7F 51 C9 FF DE D0 FC  CC
231   DATA 02 58 51 CA FF D4 41 F8 01 AE  03
232   DATA 70 0F 32 3C 00 00 74 0F 10 C0  B0
233   DATA 10 FC 00 09 20 FC FF FE 01 80  8F
234   DATA 30 C1 D0 3C 00 10 92 7C 00 00  3A
235   DATA 51 CA FF E8 20 BC FF FF FF FE  49
236   DATA 21 F8 00 6C 01 54 21 FC 00 00  71
237   DATA 01 00 00 6C 23 FC 00 00 01 7A  02
238   DATA 00 DF F0 80 43 FA 00 10 4E AE  B1
239   DATA FF A0 20 40 20 68 00 16 70 00  3E
240   DATA 4E 75 64 6F 73 2E 6C 69 62 72  85
241   DATA 61 72 79 00 48 E7 FF FE 08 39  DE
242   DATA 00 05 00 DF F0 1F 67 40 0C 38  4F
243   DATA 00 30 01 88 65 04 53 38 01 88  E5
244   DATA 2C 78 00 04 43 FA 00 46 4E AE  05
245   DATA FE DA 4A 80 67 24 21 FA 00 28  A7
246   DATA 00 6C 23 FA 00 38 00 DF F0 80  D7
247   DATA 33 FC 80 20 00 DF F0 96 20 3C  14
248   DATA 00 00 2A 80 22 7A 00 26 4E AE  B6
249   DATA FF 2E 4C DF 7F FF 4E F9 74 6F  7D
250   DATA 70 61 7A 2E 66 6F 6E 74 00 00  8C
251   DATA 01 54 00 08 00 00 43 4F 4E 00  CF
252   DATA 67 72 61 70 68 69 63 73 2E 6C  CB
253   DATA 69 62 72 61 72 79 00 96 81 80  4D
254   DATA 00 96 00 20 01 82 00 00 00 8E  47
255   DATA FF 81 00 90 3F C1 00 92 00 38  C1
256   DATA 00 94 00 D0 01 08 00 00 01 02  BA
257   DATA 00 00 01 04 00 24 00 E0 00 00  EC
258   DATA 00 E2 00 00 01 00 12 00 2C 78  85
259   DATA 00 04 93 C9 4E AE FE DA 23 C0  FF
260   DATA 00 00 00 04 28 40 4A AC 00 AC  7A
261   DATA 66 16 41 EC 00 5C 4E AE FE 80  B2
262   DATA 41 EC 00 5C 4E AE FE 8C 23 C0  36
263   DATA 00 00 00 00 43 FA FF 16 4E AE  65
264   DATA FE 68 23 C0 00 00 0E C6 43 FA  F0
265   DATA FF 7E 4E AE FE 68 23 C0 00 00  01
266   DATA 0E CA 43 FA 01 58 4E AE FE 68  02
267   DATA 23 C0 00 00 0E CE 42 B9 00 00  5E
268   DATA 0E D2 22 3C 00 00 0D A3 4B F9  E4
269   DATA 00 00 07 6C 2C 3C 00 00 4E 20  14
270   DATA 4E BA FD 2A 66 36 70 02 41 FA  DE
271   DATA 03 C6 12 1D BB FC 00 00 55 8C  4C
272   DATA 67 26 08 81 00 05 B2 18 66 E8  41
273   DATA 51 C8 FF EC 70 11 1A D8 51 C8  54
274   DATA FF FC 22 3C 00 00 0D B6 4B F9  C7
275   DATA 00 00 07 6C 4E BA FD 08 67 0C  FC
276   DATA 41 FA 03 D4 4E BA FC 72 60 00  58
277   DATA FB 0E 2C 7A 04 C6 2A 6E 00 38  16
278   DATA 4B ED 00 2C 23 CD 00 00 0E BA  2A
279   DATA 2A 6D 00 04 2A 6D 00 04 2C 15  09
280   DATA 23 C6 00 00 0E B6 4E BA FB 2C  CE
281   DATA 70 0F 43 FA 01 04 11 59 00 05  E6
282   DATA 11 59 00 07 42 81 12 19 31 41  D2
283   DATA 00 08 53 41 31 41 00 40 31 41  E6
284   DATA 00 44 31 7C 00 0B 00 0A 31 7C  4A
285   DATA 00 0A 00 46 31 7C 00 0A 00 4A  5A
286   DATA 56 68 00 0E B0 7C 00 0A 65 04  D9
287   DATA 56 68 00 0C 52 68 00 10 31 47  7E
288   DATA 00 26 21 7C 01 00 00 05 00 30  CC
289   DATA D0 FC 00 2C 21 48 FF E6 D0 FC  3F
290   DATA 00 10 21 48 FF F8 D0 FC 00 14  E8
291   DATA 21 48 FF B0 52 47 51 C8 FF 9A  47
292   DATA 42 A8 FF B0 54 68 FF B6 55 68  3D
293   DATA FF DE 54 68 FF B4 55 68 FF DC  D1
294   DATA 56 68 FF C0 21 48 FF D2 31 7C  B8
295   DATA 00 29 00 0A D0 FC 00 24 21 48  B2
296   DATA FF DC 41 FA 00 3E 4E AE FF 34  8F
```

```
297   DATA 23 C0 00 00 0E BE 67 00 FA 3E  8D
298   DATA 20 40 23 E8 00 32 00 00 0E C2  11
299   DATA 4B FA 00 82 70 1E 4E BA FB 10  BB
300   DATA 51 C8 FF FA 60 00 F5 14 69 6E  22
301   DATA 74 75 69 74 69 6F 6E 2E 6C 69  95
302   DATA 62 72 61 72 79 00 00 B4 00 00  5C
303   DATA 01 18 00 C8 00 01 00 08 07 48  D5
304   DATA 00 01 10 0E 00 00 00 62 00 00  AA
305   DATA 00 00 00 00 0D 47 00 00 00 00  1C
306   DATA 00 00 00 00 00 64 00 64 00 64  92
307   DATA 00 64 00 01 6D 4B 0C 85 4B 0C  79
308   DATA 9D 4B 0C 5D 27 5C AA 61 2C AA  C8
309   DATA 6D 2C 7D 0B 1C 0E 94 7C 8E 94  2F
310   DATA 7C 0E 7C FC 0E 88 FC 0E A0 7C  74
311   DATA 8E A0 7C E6 AF 24 E6 BB 24 5E  66
312   DATA AF 84 04 13 48 69 6E 74 65 72  C8
313   DATA 67 72 75 6E 64 00 7F 13 3C 2D  7E
314   DATA 3E 00 BA 13 20 54 65 78 74 66  1D
315   DATA 61 72 62 65 20 00 14 5D 52 00  30
316   DATA 2C 5D 47 00 44 5D 42 00 CB 5D  0C
317   DATA 52 00 E3 5D 47 00 FB 5D 42 00  26
318   DATA 57 24 41 62 73 74 75 66 75 6E  EB
319   DATA 67 20 61 75 66 00 5F 40 52 47  1C
320   DATA 42 2D 41 75 73 77 61 68 6C 00  2F
321   DATA 57 48 66 FC 72 20 41 62 73 74  AE
322   DATA 75 66 75 6E 67 00 6F 53 52 00  20
323   DATA 87 53 47 00 9F 53 42 00 42 69  8C
324   DATA 41 62 73 63 68 61 6C 74 75 6E  79
325   DATA 67 3A 00 42 75 53 63 72 6F 6C  C7
326   DATA 6C 69 6E 67 3A 00 48 84 42 6F  4C
327   DATA 6F 74 2D 54 65 78 74 20 61 6E  1F
328   DATA 73 65 68 65 6E 00 18 90 42 6F  47
329   DATA 6F 74 2D 42 6C 6F 63 6B 20 61  DC
330   DATA 75 66 20 64 66 30 3A 20 73 63  18
331   DATA 68 72 65 69 62 65 6E 00 14 9C  80
332   DATA 54 65 78 74 20 65 64 69 74 69  D2
333   DATA 65 72 65 6E 00 9C 9C 54 65 78  46
334   DATA 74 20 6C F6 73 63 68 65 6E 00  8D
335   DATA 24 A8 54 65 78 74 20 6C 61 64  F4
336   DATA 65 6E 00 94 A8 54 65 78 74 20  F8
337   DATA 73 70 65 69 63 68 65 72 6E 00  07
338   DATA 10 B7 54 65 78 74 6E 61 6D 65  41
339   DATA 3A 00 EC B7 43 4C 49 00 E8 C3  09
340   DATA 49 6E 66 6F 00 20 C3 53 74 61  F2
341   DATA 74 75 73 3A 00 60 C3 4F 4B 21  EE
342   DATA 00 5F 2F 48 69 6E 74 65 72 67  BE
343   DATA 72 75 6E 64 00 B0 69 61 75 74  39
344   DATA 6F 00 B4 75 65 69 6E 00 5F 2F  55
345   DATA 20 54 65 78 74 66 61 72 62 65  73
346   DATA 20 00 AC 69 20 43 4C 49 20 00  D0
347   DATA B4 75 61 75 73 00 5F C3 20 20  40
348   DATA 20 20 20 20 20 20 20 20 20 20  3C
349   DATA 20 20 00 60 C3 44 4F 53 2D 46  B6
350   DATA 65 68 6C 65 72 21 00 60 C3 54  8E
351   DATA 65 78 74 6E 61 6D 65 3F 00 60  B6
352   DATA C3 4B 65 69 6E 20 54 65 78 74  A6
353   DATA 21 00 60 C3 53 63 68 6E 69 70  F8
354   DATA 70 21 00 60 C3 42 6F 6F 74 42  18
355   DATA 6C 6F 63 6B 2E 2E 2E 00 60 C3  BC
356   DATA 63 2F 42 6F 6F 74 4F 66 66 2E  45
357   DATA 2E 2E 00 60 C3 44 69 73 6B 2D  D2
358   DATA 46 65 68 6C 65 72 21 00 60 C3  E8
359   DATA 53 63 68 72 65 69 62 73 63 68  C0
360   DATA 75 74 7A 3F 00 42 6F 6F 74 4C  48
361   DATA 65 74 74 65 72 20 31 2E 30 20  57
362   DATA 20 20 A9 62 76 72 00 74 72 61  B3
363   DATA 63 6B 64 69 73 6B 2E 64 65 76  C0
364   DATA 69 63 65 00 4E 49 4C 3A 00 6E  36
365   DATA 65 77 63 6C 69 20 63 6F 6E 3A  B5
366   DATA 30 2F 30 2F 36 34 30 2F 31 35  21
367   DATA 39 2F 43 4C 49 00 52 41 57 3A  05
368   DATA 31 34 35 2F 30 2F 33 35 30 2F  01
369   DATA 31 34 31 2F 45 64 00 53 59 53  01
370   DATA 3A 63 2F 65 64 00 63 2F 42 6F  5C
371   DATA 6F 74 4F 66 66 00 72 61 6D 3A  8C
372   DATA 65 64 00 72 61 6D 3A 65 64 20  5E
373   DATA 72 61 6D 3A 62 6F 6F 74 2E 74  2B
374   DATA 78 00 00 F8 12 45 64 20 6D 61  21
375   DATA 63 68 74 20 50 72 6F 62 6C 65  99
376   DATA 6D 65 21 20 00 00 00 FC 0E 54  38
377   DATA 65 78 74 20 67 65 73 70 65 69  57
378   DATA 63 68 65 72 74 3F 00 01 00 10  0A
379   DATA 16 6C 69 6E 6B 65 20 4D 61 75  14
380   DATA 73 74 61 73 74 65 3A 20 4A 41  22
381   DATA 00 01 01 A8 16 72 65 63 68 74  47
382   DATA 65 20 4D 61 75 73 74 61 73 74  50
383   DATA 65 3A 20 41 42 42 52 55 43 48  A3
384   DATA 00 00 00 6C 12 A9 20 31 39 38  19
385   DATA 38 20 20 42 2E 76 61 6E 20 52  76
386   DATA FC 73 63 68 65 6E 2C 20 4E 6F  02
387   DATA 72 64 65 72 73 74 72 2E 33 34  0C
388   DATA 2C 20 32 39 31 30 20 57 65 73  22
389   DATA 74 65 72 73 74 65 64 65 00 00  6B
390   DATA 00 00 03 F3 00 00 00 00 00 00  5B
391   DATA 00 01 00 00 00 00 00 00 00 00  89
392   DATA 00 00 00 05 00 00 03 E9 00 00  F9
393   DATA 00 05 31 FC 4E 71 01 28 0C B8  25
394   DATA 00 00 01 00 00 6C 67 F6 4E 75  E6
395   DATA 00 00 00 00 03 F2 00 00 03 F2  D5
396   DATA 00 00 00 00 00 00 00 00 00 FF  82
397   DATA 00 00 00 00 00 00 00 00 00 00  8D
398   DATA 00 00 00 00 00 00 00 00 00 00  8E
399   DATA 00 00 00 00 00 00 00 00 00 00  8F
400   DATA 00 00 00 00 00 00 00 00 03 EC  E3
401   DATA 00 00 00 41 00 00 00 00 00 00  95
402   DATA 00 46 00 00 00 5C 00 00 00 64  2E
403   DATA 00 00 00 6E 00 00 00 C6 00 00  7B
404   DATA 00 CC 00 00 00 D4 00 00 00 E2  F8
405   DATA 00 00 00 F0 00 00 00 FE 00 00  45
406   DATA 01 0C 00 00 01 1A 00 00 01 28  E9
407   DATA 00 00 01 76 00 00 01 94 00 00  19
408   DATA 01 F4 00 00 02 56 00 00 02 5A  25
409   DATA 00 00 02 68 00 00 02 6C 00 00  AD
410   DATA 02 88 00 00 02 C0 00 00 02 DC  E0
411   DATA 00 00 03 00 00 00 03 06 00 00  E9
412   DATA 03 10 00 00 03 1A 00 00 03 36  A1
413   DATA 00 00 03 5E 00 00 03 62 00 00  43
414   DATA 03 7E 00 00 03 8A 00 00 03 A6  7F
415   DATA 00 00 03 C2 00 00 03 CA 00 00  15
416   DATA 03 F0 00 00 03 F4 00 00 04 38  9E
417   DATA 00 00 04 64 00 00 04 6C 00 00  B9
418   DATA 04 8E 00 00 04 B0 00 00 04 B6  36
419   DATA 00 00 05 18 00 00 06 98 00 00  FC
420   DATA 06 9C 00 00 07 22 00 00 07 44  B8
421   DATA 00 00 07 48 00 00 07 52 00 00  9B
422   DATA 07 5A 00 00 07 5E 00 00 07 66  F3
423   DATA 00 00 07 74 00 00 09 86 00 00  FB
424   DATA 09 94 00 00 09 A2 00 00 09 A8  B3
425   DATA 00 00 09 AE 00 00 09 EA 00 00  0B
426   DATA 0A 14 00 00 0A 24 00 00 0A CE  4C
427   DATA 00 00 0A DC 00 00 0B 1E 00 00  76
428   DATA 00 1B 00 00 00 01 00 00 00 06  24
429   DATA 00 00 01 9A 00 00 01 A0 00 00  1F
430   DATA 01 AC 00 00 01 C4 00 00 01 CA  91
431   DATA 00 00 01 E0 00 00 02 0C 00 00  A0
432   DATA 02 26 00 00 02 3E 00 00 02 48  5E
433   DATA 00 00 02 D6 00 00 03 AC 00 00  84
434   DATA 05 3E 00 00 05 4A 00 00 05 5A  B9
435   DATA 00 00 05 D6 00 00 06 90 00 00  C4
436   DATA 06 B8 00 00 06 D2 00 00 07 92  27
437   DATA 00 00 09 5A 00 00 09 78 00 00  37
438   DATA 09 B4 00 00 09 CE 00 00 09 F0  D9
439   DATA 00 00 0B 16 00 00 00 00 00 00  30
440   DATA 03 F2 00 00 03 EB 00 00 15 63  CB
441   DATA 00 00 03 F2                    8A#
```

Listing 2: Für Tippfreudige, die nicht auf die Programmdiskette der Redaktion warten wollen, haben wir hier den Basiclader zu BOOTLETTER abgedruckt.

DER GROSSE KICKSTART

HARDWARE WETTBEWERB

Der Amiga ist ein offenes System, es bietet reichlich Schnittstellen zur Außenwelt. Nach den Standardschnittstellen, wie RS 232 und Druckerport, sind RGB Port und der Expansionport die interessantesten Verbindungen für Peripheriegeräte. In der Theorie mag das den Anwender begeistern, auf der praktischen Seite fehlen jedoch einfach die entsprechenden Geräte und Erweiterungen. Wir wollen genau da ansetzen und mit diesem Wettbewerb Abhilfe schaffen.

Wer kann mitmachen ?

Es ist jede Person an unserem Hardwarewettbewerb beteiligt, die bis zum Einsendeschluß einen fertigen Prototyp und die Schaltungsbeschreibung einsendet. Die Schaltung muß natürlich frei von Rechten Dritter und darf bis jetzt noch nicht in einer anderen Art veröffentlicht worden sein.

Was darf es sein ? Gefragt sind bei dieser Ausschreibung alle Arten von Hardware. Als Anregung hier eine kleine Liste mit Vorschlägen :

- Speichererweiterungen
- MIDI-Schaltungen
- Stereo-Verstärker
- Schnittstellentreiber
- Programmiergeräte (z.B. PAL Brenner)
- Video oder HF Modulatoren
- Soundsampler
- Videodigitizer
- Meßwerterfassung
- Ein/Ausgabe Port
- Soundsampler
- Digitizer
- Meßwerterfassung
- Ein/Ausgabe-Port

oder was Ihnen hier sonst noch einfält

Der Phantasie der Entwickler sollen hier keine Grenzen gesetzt werden. Auch kleinere Projekte werden hier berücksichtigt, zumal auch der Bauteileaufwand sowie die Kosten der Schaltung in die Bewertung mit eingehen. Das Speeder-Projekt in dieser Ausgabe gibt Ihnen ein weiteres Beispiel für -wie könnte es aussehen-.

Das müssen Sie tun. Schicken Sie Ihre Schaltung mit Beschreibung und Prototyp an folgende Adresse :

MERLIN Computer Gmbh
Kennwort Kickstart-Hardwarewettbewerb
Industriestr. 26, 6236 ESCHBORN

Der Einsendeschluß ist der 16.6.88 , es gilt das Datum des Poststempels.

Die Gewinne

Wir haben Preise im Wert von 6600,- DM ausgesetzt.

1. Preis 3000,- DM

2. Preis 2000,- DM

3. - 10. Preis je 20 PD Disketten nach freier Auswahl aus unserem Angebot

Teilnahmebedingungen:

Die Vergabe der Preise erfolgt durch eine aus Redaktionsmitgliedern gebildete Jury. Der Rechtsweg ist ausgeschlossen.

Alle eingesandten Schaltungen, Schaltpläne und Prototypen werden nach ihrer Beurteilung durch die Jury an die Einsender zurückgeschickt.

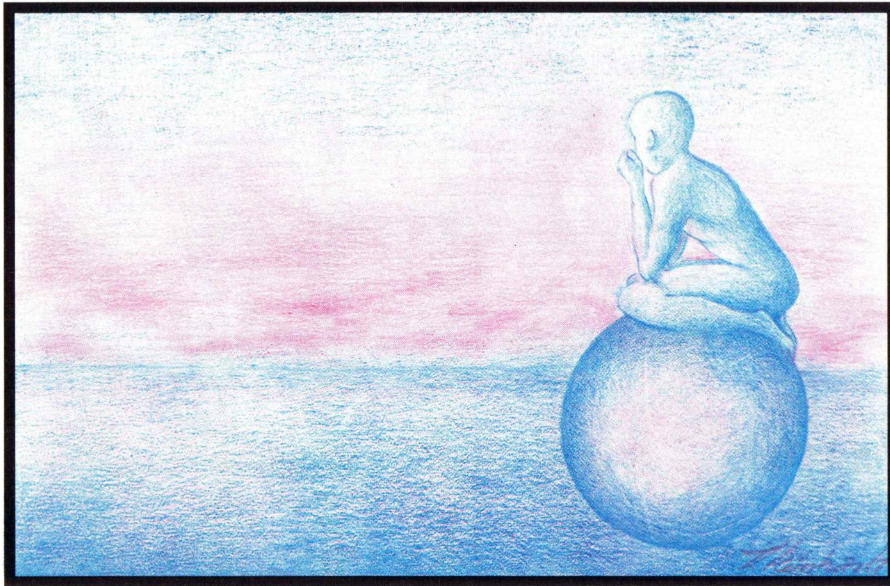
Der Teilnehmer gibt durch seine Einsendung die Erklärung ab, daß die Schaltung frei von Rechten Dritter ist. Das Copyright der Preise 1. und 2. geht an die Merlin Computer Gmbh. Bei allen übrigen Einsendungen bleibt das Copyright beim jeweiligen Autor. Sollten unter diesen Projekten für eine Veröffentlichung intessante Vorschläge sein, dann werden wir uns mit dem Autor in Verbindung setzen, um ihm ein Angebot zu machen.

Mitarbeitern der Merlin Computer Gmbh und deren Angehörigen ist die Teilnahme untersagt.

VON JOBST HERMEYER

LISP!

Dialekt der Experten



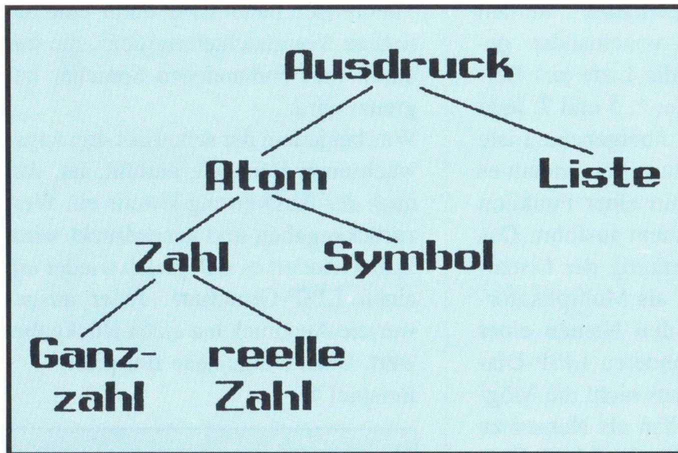
Beim Durchforsten des Softwareangebots für den Amiga fällt des öfteren die Programmiersprache LISP ins Auge. Da die wenigsten Amiga-User mit diesem Stichwort etwas anfangen können, soll hier einmal ein bißchen Licht in die dunklen LISP-Katakomben gebracht werden.

Vorgeschichte

Mit dem Jahr 1985 begann eine AI-Welle (AI = Artificial Intelligence = Künstliche Intelligenz) den Computermarkt zu überfluten. Schlagwörter wie Expertensysteme, "PROLOG" und "natürlichsprachlich" zierten die Titel vieler einschlägiger Magazine; eilig begonnene Zeitschriftenserien und hastig gefüllte Bücher überschwemmten den Markt. Ziel dieser Welle war häufig die semiprofessionelle Welt der Personal Computer und die Besitzer von Home Computern, deren Geräte durch die expontential steigende Leistungsfähigkeit dem Bereich ernsthafter Anwendungen immer näher rücken.

Doch so heftig die Brandung tobte, so schnell ebbte sie auch wieder ab. Das Ergebnis ist, daß viele Leute inzwischen "mal etwas" von künstlicher Intelligenz gehört haben und trotzdem keiner so richtig weiß, was man damit anfangen kann.

Natürlich ist auch einiges Positive zurückgeblieben. Da wäre zuerst das Angebot an spezifischen Programmiersprachen wie PROLOG oder eben LISP, die nun zu erschwinglichen Preisen für Microcomputer zur Verfügung stehen. Dann folgen die ganzen Endprodukte, wie fertige Expertensysteme, sogenannte Shells (das sind quasi leere Expertensysteme, die vom Benutzer auf seine spezifische Problemstellung adaptiert werden können) und Hardwareerweiterungen, die etwa einen einfachen PC zur 8 MB LISP - Maschine machen. Solche Leckerbissen sind aber bisher leider auf die MS-DOS Welt beschränkt, da es dort offensichtlich genug zahlungskräftige Kundschaft gibt. Übriggeblieben ist auch in jeder guten Fachbuchhandlung die extra eingerichtete Rubrik künstliche Intelligenz, vollgestopft mit jeder nur denkbaren Literatur um und an dem Thema. Für den "Amiga-Normalverbraucher" hat diese Bücherecke aber so ihre Nachteile. Da das Herz der AI nach wie vor in den USA schlägt sind die meisten Werke in Englisch abgefaßt und nur die wirklich verbreiteten Standardausgaben liegen in deutscher Übersetzung vor. Zudem sind die, meist in kleiner Auflage, direkt importierten Bücher aber auch die Werke deutscher Autoren und Verlage mit 70.- bis über 100.- DM für schmale Börsen ziemlich unerschwinglich. Bleibt noch, daß manche Sachen für den "Amiganormalverbraucher" einfach zu schwierig sind, da sie sich ausschließlich an die Adresse von Experten und Informatikstudenten höherer Semester richten.



LISP-Objekte

Das reicht auch schon an Gründen, um ein bißchen AI in der KICKSTART zu betreiben. Aber Achtung! Ziel des Artikels ist es nicht, daß die Leser nach der Lektüre Projekte in fließendem LISP abwickeln können; vielmehr sollte man eine gute Idee davon bekommen, wie LISP funktioniert und was man damit normalerweise macht, beziehungsweise machen kann. Es wäre auch erfreulich, wenn sich das Interesse einiger Amigafreaks für diesen Expertendialekt wecken ließe. Nebenbei sollen deshalb auch hier und dort einige allgemeine Informationen über das Fachgebiet eingestreut werden.

Zur Sache

Erst einmal zu der Frage, ob sich AI und LISP wirklich so in einem Atemzug erwähnen lassen, wie das bis eben geschah. Die Antwort darauf kann definitiv ja lauten, wofür gewichtige historische Gründe vorliegen.

Man versetze sich zurück in den Sommer des Jahres 1956 (nein, das ist kein Druckfehler, wir reden tatsächlich über 1956). John McCarthy, ein junger Assistenzprofessor für Mathematik am Dartmouth College, und sein Freund Marvin Minsky vom berühmten Massachusetts Institute of Technology (MIT) beschließen, eine Konferenz über "künstliche Intelligenz" abzuhalten. Auf dieser als "Dartmouth Conference" in die Computergeschichte eingegangenen Veranstaltung ist die Benutzung des Terms künstliche Intelligenz zum erstenmal schriftlich verbürgt. Hier wurde der Grundstein für ein neues Fach der Computerwissenschaft gelegt. Und es war auch hier, daß eben jener John McCarthy die Anregung dazu erhielt eine Algorithmik zu entwickeln, mit deren Hilfe man zum Beispiel Probleme der symbolischen Algebra kodieren kann. LIS(ten) P(rogrammierung) war das Ergebnis.

Sprachgut

Hoffentlich haben sich inzwischen nicht schon einige gelangweilte Leser ausgeklinkt, es wird vorerst noch etwas trockener. Die Rede soll nämlich von den LISP-Dialekten sein. LISP ist wahrscheinlich die Programmiersprache mit den meisten unterschiedlichen Implementationen überhaupt. Um einen Eindruck davon zu bekommen, stelle man sich einen, na sagen wir, Chinesen mit leidlichen Deutschkenntnissen vor, der sich unvermittelt auf einem "Wettbewerb für deutsche Volkstanzgruppen" wiederfindet. Bei all den unterschiedlichen Sprachdialekten, die er dort zu hören bekommt, wird er am Ende vielleicht zu der Überzeugung gelangen, es gäbe gar keine eigentliche deutsche Sprache. Ähnlich verhält es sich mit LISP. Die Tatsache, daß ihr Einsatz während der ganzen Zeit auf Universitäten und Institute beschränkt blieb, trug im wesentlichen dazu bei, daß sich immer wieder neue Abarten bilden konnten.

Berühmte LISP-Ableger, die eigene Namen erhielten, sind zum Beispiel LOGO, PLANNER oder das hochaktuelle SCHEME, daß jeder MIT-Anfänger als erste Programmiersprache zu lernen hat. Unter dem, was sich weiterhin LISP nennt ist die Auswahl auch ziemlich groß. Da gibt es MACLISP, MULISP, LISP Company LISP, INTERLISP, Cambridge LISP oder Common LISP und einige mehr. Die letzten beiden Abarten sind seit geraumer Zeit auch für den Amiga erhältlich. Cambrigde LISP liegt in einer recht professionellen Version mit Compiler von Metacomco vor und wird in einer der nächsten Ausgaben getestet. Common LISP scheint sich langsam zum heimlichen Standard zu mausern, da es einerseits vom Sprachschatz her auf kommerzielle Systeme ausgerichtet ist, andererseits in immer mehr Lehrbüchern verwendet wird und auf vielen Computersystemen in einer lauffähigen Version existiert. Letzteres ist zu nicht unwesentlichem Teil XLISP (schon wieder ein Dialekt) zuzuschreiben , einer Common LISP-Implementation, die für die Beispiele im Artikel verwendet wird. XLISP ist in C geschrieben, deshalb sehr portabel und außerdem Public Domain. Die Amigaversion ist auf Diskette 17 des PD-Service der KICKSTART-Redaktion zu erhalten, so daß jeder die Beispiele fast kostenlos auf seinem eigenen LISP nachvollziehen kann. Also schleunigst die beiliegende Bestellkarte ausfüllen und abschicken!

Und was kann man diesem Wirrwarr der Dialekte nun entnehmen? Daß LISP etwa eine Sprache für Anarchisten sei? Beileibe nicht! Bei normierten Sprachen wie PASCAL oder FORTRAN wirken nachträgliche Erweiterungen oft aufgesetzt und sprachfremd, sind jedoch bitter notwendig, um die Sprachen gegenüber Neuerungen in Hard- und Software-Konzepten nicht hoffnungslos antiquieren zu lassen. Im Gegensatz dazu ist die Bildung vom Sprachdialekten ein Teil der Entwicklungsgeschichte von LISP. Was zum Beispiel bei BASIC zu einem heillosen Chaos der Abarten führte wirkt bei LISP durchaus produktivitätssteigernd. Die Dialekte haben die Sprache über all die Jahre jung erhalten und die Brauchbarkeit des Sprachkerns, beziehungsweise der sich dahinter verbergenden Philosophie bestätigt.

Na endlich

Mit einer der besten Wege, eine Programmiersprache zu erlernen, ist einfach ins kalte Wasser zu springen und sich ein Beispiel anzuschauen. Beispiel 1:

```
> (* 5 7)
35
```

Diese zwei Zeilen sagen schon eine ganze Menge über die Funktionsweise von LISP aus. Da ist zuerst das > Zeichen. Das ist der Prompt (das "Bereitzeichen"), vergleichbar mit dem allgemein von BASIC bekannten OK oder READY. Das LISP-System signalisiert damit seine Bereitschaft, Konsoleneingaben zu empfangen und auszuwerten. Und hier liegt die große und ziemlich einzige Gemeinsamkeit mit BASIC. LISP ist genauso wie letzteres, vorzugsweise als interpretative Umgebungssprache implementiert. Das heißt die Eingaben einer Zeile werden direkt nach Betätigung eines Zeilenvorschubs (bei manchen LISPs auch ohne diesen) ausgewertet und das Ergebnis wird ausgedruckt. Dieser Vorgang (Eingabe, Verarbeitung, Ausgabe) stellt eines der Grundprinzipien der Funktionsweise der elektronischen Datenverabeitung dar. Er ist auch das Fundament eines jeden LISP-Systems; hier heißt dieser Zyklus READ-EVAL(uate)-PRINT-LOOP.

Das nächste Zeichen gehört auch zu den Grundelementen der Sprache:

die Klammer. An dieser Stelle gleich ein offener Ratschlag: wer mit Klammern - hauptsächlich runden - auf Kriegsfuß steht, sollte sich lieber zweimal überlegen, ob er mit LISP "etwas anfängt". Selbst bei einfachen Ausdrücken (etwa zehn Zeilen) sind dutzende von Klammerebenen keine Besonderheit. Diese Tatsache veranlaßt Kritiker immer wieder zu Schmähungen wie der, daß LISP-Programmierer schon an Klammern erstickt seien. Die eingefleischten Fans der Sprache kontern solche Behauptungen lächelnd mit dem Vorwurf des fehlenden (Ein-) Überblicks ihrer Gegner.

Und nun zur Bedeutung der Klammern. Ein Klammernpaar stellt immer eine Liste dar. Zwei Klammern ohne Inhalt sind folgerichtig eine leere Liste. Der Inhalt einer Liste sind ihre Elemente; zwei Elemente werden durch Leerzeichen voneinander getrennt. Danach hat die Liste aus Beispiel 1 drei Elemente: *, 5 und 7. Jede dem System direkt übergebene Liste versucht dieses auszuwerten, indem es das erste Element mit einer Funktion identifiziert, die es dann ausführt. Das bringt uns zur Bedeutung der Listenelemente. Der Stern als Multiplikatorzeichen stellt also den Namen einer Funktion dar. Bei anderen LISP-Dialekten gibt es übrigens nicht die Möglichkeit, Sonderzeichen als Namen zu verwenden, dort heißt die Multiplikation meistens MUL. Die beiden Zahlen repräsentieren die Argumente der Funktion *. In LISP wird also von der Prefixnotation Gebrauch gemacht, bei der zuerst die Funktion und dann die Parameter genannt werden. Der Vorteil hiervon wird schnell an einem weiteren Beispiel klar. Beispiel 2:

```
> (+ 10 7 15 9)
40
```

Bei der Prefixnotation ist es nämlich möglich, beliebig viele Argumente anzufügen, da die Funktion ja am Anfang genannt wird. Hat man aber eine Ausdruck mit verschiedenen Operatoren (Funktionen), sieht man, wie LISP die Komplexität rapide anwachsen läßt. Beispiel 3:

```
> (+ 30 (- (* 16 2) 19))
43
```

In BASIC würde dieser Ausdruck, als 30+16*2-19 geschrieben, viel einfacher ausssehen. Aber erstens kann Komplexität keinen LISP-Fan erschrecken, und zweitens ist in vielen Fällen die strenge Systematik dieser Vorgehensweise eine nützliche Hilfe auf dem Weg zur Problemlösung. Außerdem kann man LISP-Ausdrükken schon durch die Schreibweise zu mehr Übersicht verhelfen.

```
(+ 30
   (- (* 16 2)
      19))
```

In dieser Weise strukturiert, wird die Sache schon etwas klarer. Beispiel 3 kann man übrigens entnehmen, daß Listen, wie etwa (* 16 2), auch selbst Elemente von Listen sein können. Theoretisch bietet LISP dafür eine beliebige Verschachtelungstiefe, die nur durch den vorhandenen Speicher begrenzt wird.

Was bei jedem der schon auf drei angewachsenen Beispiele auffällt, ist, daß nach der Auswertung jeweils ein Wert zurückgegeben und ausgedruckt wird. Dabei handelt es sich gleich wieder um einen LISP-Grundsatz. Jeder ausgewertete Ausdruck hat einen Rückgabewert. Dazu einige neue Beispiele. Beispiel 4:

```
> ()
NIL
> 23
23
> '(* 7 5)
(* 7 5)
```

Nichts, der Inhalt der leeren Menge, wird offensichtlich durch NIL, ein LISP-Atom, repräsentiert, auf das wir später noch stoßen werden. Zahlen sind ebenfalls Atome und stehen für sich selbst. Der dritte Term scheint auf Anhieb etwas merkwürdig zu sein. Offensichtlich hat hierbei keine Auswertung stattgefunden; das EVAL in der READ-EVAL-PRINT-Schleife scheint zu fehlen, wenn da nicht dieses kleine Hochkomma wäre. Das Hochkomma - im englischen Quote - ist eigentlich nur eine Abkürzung für die Funktion (QUOTE). "To quote" heißt zitieren, und genau das macht die Funktion auch. Beispiel 5:

```
> ''(* 7 5)
(QUOTE(* 7 5))
```

QUOTE scheint allerdings nicht nur zu zitieren, sondern auch Abkürzungen auszuschreiben. Abgesehen von dieser Ausnahme zeigt Beispiel 5, daß ' und (QUOTE) equivalent sind und daß alles zitiert wird, was hinter einem Anführungszeichen steht. Und was kann man damit anfangen? Ein wichtiges Anwendungsgebiet des QUOTEs ist die Zuweisung von Werten zu Platzhaltern respektive Variablen. Allerdings verwendet man statt Variable in LISP eher den Begriff Symbol, da diesem auch eine etwas andere Bedeutung zukommt. Die eingebaute Funktion (SETQ) wird meistens für die Symbolzuweisung verwendet. Beispiel 6:

```
>(SETQ X (* 20 7))
140

>X
140

>(SETQ A (+ X 50))
190

>(SETQ X '(* 20 7))
(* 20 7)

>X
(* 20 7)
```

SETQ gibt, wie oben gefordert, auch brav immer einen Wert zurück. Außerdem bleibt der Wert der zugewiesenen Symbole X und A über die SETQ-Operation hinaus erhalten. Man spricht davon, daß SETQ eine Funktion mit Nebenwirkung ist. Das Q in SETQ bedeuted, wie manche vielleicht schon vermuteten, nichts anderes als QUOTE und heißt, daß das erste Argument der Funktion nicht ausgewertet wird.

```
(SETQ X (* 20 7))
```

bewirkt deshalb das Gleiche wie

```
(SET (QUOTE X) (* 20 7))
```

nur daß man das QUOTE spart.

Zurück zu Beispiel 6. In zwei Fällen wird das Symbol X mit einem Wert belegt. Das erstemal, als Ergebnis eines evaluierten Ausdrucks, mit der Zahl 140, einem LISP-Atom. Im unteren Teil des Beispiels wird X mit einer Liste, (* 20 7), identifiziert. Zwischendurch bekommt noch das Symbol A einen Wert zugewiesen. An der Zuweisung von A läßt sich zeigen, daß prinzipiell kein Unterschied zwischen einem Symbol und seinem Wert besteht. Mit dieser Feststellung sind wir nun auf einen der Kernpunkte der LISP-Philosophie gestoßen: die Gleichbehandlung von Daten und Programmkode. In den meisten anderen Programmiersprachen gibt es eine klare Trennung zwischen Daten- beziehungsweise Variablenbereichen und ausführbaren Programmteilen. Am stärksten ist diese Trennung wohl beim guten alten COBOL ausgeprägt, bei dem es eine streng getrennte DATA DIVISION und eine PROCEDURE DIVISION gibt. Bei den klassischen Programmiersprachen verwischt C die Trennung wahrscheinlich noch am meisten, indem es Zeiger auf Funktionen zuläßt. Keiner der Kandidaten hat aber eine explizite Empfehlung zur Aufhebung des Unterschieds im Sprachkonzept.

Um die Bedeutung der Gleichbehandlung von Steuersequenzen und Daten ermessen zu können, ist es ratsam, einen kleinen Exkurs in die Geschichte der Mathematik zu unternehmen.

Gegen Ende des 19. Jahrhunderts befand sich die Mathematik in einer Art Krise. Immer mehr neue Teiltheorien, wie zum Bespiel die Mengenlehre von Cantor, entstanden. Aber niemand konnte auf Anhieb sagen, ob sie wirklich "wohldefiniert" waren und wie sie sich in das Gesamtgebäude der Mathematik einfügen sollten. Es herrschte ein allgemeines Defizit an mathematischem Beweis. Nun ist die Jahrhundertwende allgemein als Zeit der Superlativen, von Brehms' Tierleben bis hin zur Titanic, bekannt. Eins dieser endgültigen Standardwerke sollten auch die "Principia Mathematika" werden. Sie sollten mit Beweiszweifeln ein für allemal aufräumen, indem sie die Mathematik als vollständig bewiesene, beziehungsweise beweisbare Theorie festschrieben. Doch wie die Titanic erwiesen, sich auch die "Principia Mathematika" als sinkbar. Der symbolische Eisberg für letztere war ein Mathematiker namens Kurt Gödel. Gödel behauptete, daß "alle wiederspruchsfreien axiomatischen Formulierungen der Zahlentheorie unentscheidbare Aussagen enthalten" [1]. Aus Platzgründen soll an dieser Stelle nicht weiter auf den Gehalt dieser 1931 veröffentlichten These eingegangen werden. Für unsere Zwecke reicht es auch aus zu verstehen, daß Gödel der absoluten Beweisbarkeit mathematischer Aussagen widersprechen wollte. Interessanter für die Analogie zu dem LISP-Grundsatz ist die Technik, wie Gödel zu der Behauptung des Satzes gekommen ist. Betrachtet werden zwei Teile der Zahlentheorie : ganze Zahlen und zahlentheoretische Aussagen. Zahlen stehen nur für sich, sind also keine Aussagen. Aussagen dagegen, sind "nur" solche und können selbst nichts über Aussagen der Zahlentheorie besagen. Gödels ebenso einfacher wie genialer Einfall besteht darin, zahlentheoretische Aussagen durch Zahlen repräsentieren zu lassen, also zum Beispiel jeder Aussage eine Kodenummer zuzuweisen. Durch diese Aktion wird es möglich, den zum strengen Beweis notwendigen Unterschied zwischen Zahlen und Aussagen zu unterminieren und den oben zitierten Satz zu formulieren. Obwohl sie auf Anhieb unscheinbar wirken mag, hat diese These die mathematische Beweis- und Logiktheorie entscheidend verändert. Einige Jahre später entwarf Alan Turing, Mathematiker und ein brillianter Vordenker der heutigen Computerwissenschaft, das Modell einer Maschine die es ermöglicht, Gödels Theorie nachzuvollziehen. In diesem rein theoretischen Konzept wird der gödelsche Beweis der Unentscheidbarkeit angewandt, der eben auch die Gleichbehandlung von Daten (Zahlen) und Steuersequenzen (Programmen) ermöglicht.

Damit ist die Schleife geschlossen. Am Ende dieser, hoffentlich nicht völlig unverständlichen, Abschweifungen sollte deutlich geworden sein, daß das besprochene Konzept von weitreichender Bedeutung ist. Dazu noch ein Beispiel. Beispiel 7:

```
>(SETQ X1 '(SETQ X2 '(* 7 20)))
(SETQ X2 '(* 7 20))

>(EVAL (EVAL X1))
140
```

Die verwendete EVAL-Funktion scheint der READ-EVAL-PRINT-Schleife entnommen und evaluiert jeden ihr übergebenen Term, ob Symbol oder Ausdruck. Im Beispiel 7 werden also Daten ausgewertet, die dann, als Steuersequenzen, ihrerseits Daten auswerten.

Nach all diesen Verwirrungen wird es aber höchste Zeit, einige Dinge richtigzustellen. Die Abbildung 1 sollte ein bißchen Übersicht über die verwendeten LISP-Objekte geben. In ihr fehlen aber die Funktionen. Funktionen lassen sich formal als erweiterte Ausdrücke oder Kombinationen einordnen. Eine Kombination ist eine geordnete Liste, deren erstes Element eine Funktion ist, welcher die darauffolgenden Elemente als Parameter übergeben werden. Um die Kategorie eines LISP-Objekts festzustellen, kann man sich

eine elementare Funktion zunutze machen. Beispiel 8:

```
>(SETQ X '(* 7 20))
(* 7 20)

>(ATOM 'X)
T

>(ATOM X)
NIL

>(ATOM 140)
T

>(ATOM ())
T
```

ATOM liefert zwei mögliche Werte zurück, T und NIL. NIL, dem wir schon oben begegneten, bedeutet in diesem Zusammenhang "logisch falsch". T steht folgerichtig für True, also wahr. T und NIL sind die möglichen Ergebnisse von Prädikatfunktionen.
ATOM testet also, ob ein Ausdruck ein LISP-Atom ist. Dabei schaut es, wie zu erwarten, sozusagen durch ein Symbol hindurch und bewerted seinen Inhalt, wenn das Symbol nicht "gequotet" ist.
An Predikatfunktionen gibt es in LISP noch eine ganze Menge.
Beispiel 8 :

```
>( > 5 2 )
T

>( < 70 7 )
NIL

>( AND T T )
T

>( AND T NIL )
NIL
```

Bei den ersten beiden handelt es sich offensichtlich um die Vergleichsoperatoren größer und kleiner als. AND ist eine logische Prädikatsfunktion, die nur dann T zurückgibt, wenn die beiden ihr übergebenen Ausdrücke zu T evaluieren. Dies ist natürlich nur eine kleine Auswahl aller eingebauten Funktionen (die, man erinnere sich, von LISP-Version zu LISP-Version variieren können). Wer das anfangs erwähnte XLISP schon in Händen hält kann sich die Liste der eingebauten Funktionen einmal in dem als Datei mitgelieferten Handbuch anschauen; es sind einfach zu viele, um sie alle besprechen zu können.
Damit soll es vorerst einmal genug des Neuen sein. Um das Aufgezeigte aber nicht allzu flüchtig werden zu lassen, bietet es sich an, ein wenig zu rekapitulieren :

- LISP ist eine alte Programmiersprache, eng mit dem Teilgebiet der künstlichen Intelligenz verknüpft.
- LISP ist meistens ein Interpreter
- LISP befaßt sich mit Symbolmanipulation
- LISP kennt Zahlen, Symbole, Listen und Funktionen
- LISP macht formal keinen Unterschied zwischen Auswertungsanweisungen und Daten
- QUOTE oder abgekürzt ' ist eine wichtige LISP-Funktion
- Wer lernt, mit LISP zu arbeiten, lernt auch, wie LISP arbeitet

Damit ist es hoffentlich gelungen, die Prinzipien von LISP zu verdeutlichen. Der zweite Teil dieses Berichts zeigt, wie man die eigentlichen Aufgaben von LISP bewältigt : die Manipulation von Listen und Symbolen, sowie die Erstellung von eigenen Funktionen und Programmen.
Zum Schluß noch eine kleine Aufgabe mit der man überprüfen kann ob die LISP-Grundbegiffe schon "sitzen":
Was ergibt folgende LISP-Sitzung?

```
>(SETQ A 12)
12

>(SETQ B 14)
14

>(SETQ C 20)
20

>(EVAL (SETQ F
     '(EVAL '(SET 'X (AND (< B
C) (> C A) (ATOM (* B C)))))))
?
```

LITERATUR:

[1] Douglas R. Hofstadter:
Gödel, Escher, Bach - Ein Endloses Geflochtenes Band,
Klett-Cotta : Stuttgard 1985

[2] Patrick H. Winston, Berthold K. P. Horn:
LISP,
Addison-Wesley Publishing Company, 1984

[3] Abelson, G. Sussman, J. Sussman:
Structure and Interpretation of Computer Programs,
The MIT Press : Cambridge, Massachusetts USA , 1985

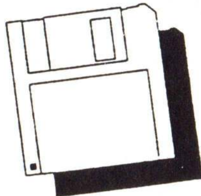
[4] F. Haugg, S. Omlor:
Lisp auf PC's
Carl Hanser Verlag : München 1987

[5] E. Charniak, D. McDermott:
Artificial Intelligence,
Addison-Wesley Publishing Company, 1985

ENDE

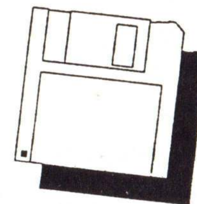
KICKSTART DISKETTEN-SERVICE

Für AMIGA-Fans, denen das Abtippen der abgedruckten Programme zu viel Mühe bedeutet, gibt es jetzt eine Alternative, denn alle in der KICKSTART abgedruckten Programm-Listings (bei Compilersprachen natürlich auch die ausführbaren Programme) befinden sich auf den KICKSTART-MONATSDISKETTEN. Auf jeder Diskette sind die Programme von zwei Monaten enthalten. Der Preis liegt bei nur DM 19.- pro Diskette zuzüglich DM 5.- Versandkosten.



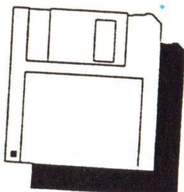
Juli/Aug. 87

Fractals
Business-Grafik
Funktions-Plotter



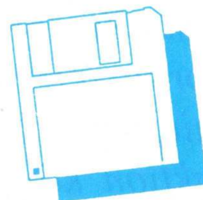
März/April 88

Röhrenschoner (Bildschirmabschalter)
Jump (IF-Befehl für's CLI)
Find (Datei-Sucher)
Fractale Landschaften (Basic)
Systemfonts in Basic
Colorhunter
DigiSound (Basic)
Viruschecker (SCA,Byte Bandit)

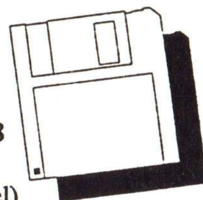


Nov./Dez. 87

MovieMaker
C-Shell
Tastenbelegung
Vokabeltrainer
Viruschecker

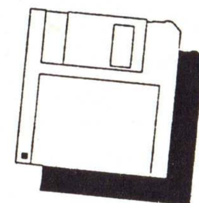


MAI / JUNI`88
- BOOTLETTER
- GURU ADE (FEHLERABFANG)
- GROSSE AUSWAHL (BATCH MENU)
- BASIC GURU (ALERT)
- GROSS GEWORDEN (DIR MIT LÄNGENANGABE)
- CURSERPOSITIONIERUNG
- AUFGERISSEN (GROSSES CLI / FENSTER)
- IFF / FORMAT / WANDLER
- SINGENDE FLOPPY
- AUTOBOOT



Jan./Feb. 88

TRON (Spiel)
KICKS (Requester, Scroll-Demo in Basic, Out, Fullscreen)
Konverter(Amiga-ASCII - PC-ASCII)
SetPCDate (Uhrzeit für PC)



Für die Bestellung können Sie die Postkarte in der Heftmitte benutzen!



VON SEBASTIAN DOSCH

DIE ZEICHENKISTE

Teil 1: Konstruktion der Perspektive

In den nächsten Ausgaben möchte ich das Thema "Perspektivisches Betrachten von Objekten und Flächen" behandeln und Ihnen einen Einblick in die Effekte geben, die mit Hilfe von perspektivischen Konstruktionen erzielt werden können. Dabei stütze ich mich auf keine besondere Software, Sie können diese Informationen auch in Ihrem selbstgeschriebenen Basic-Zeichenprogramm benutzen. Die einzige Voraussetzung an die Software ist, daß sie eine Linienfunktion enthalten muß. Auch eine Kreisfunktion könnte in dieser Ausgabe nicht schaden. Aber fangen wir diesmal mit ein paar leichten und grundlegenden Übungen zu diesem Thema an.

Die Konstruktion der perspektivischen Frontansicht

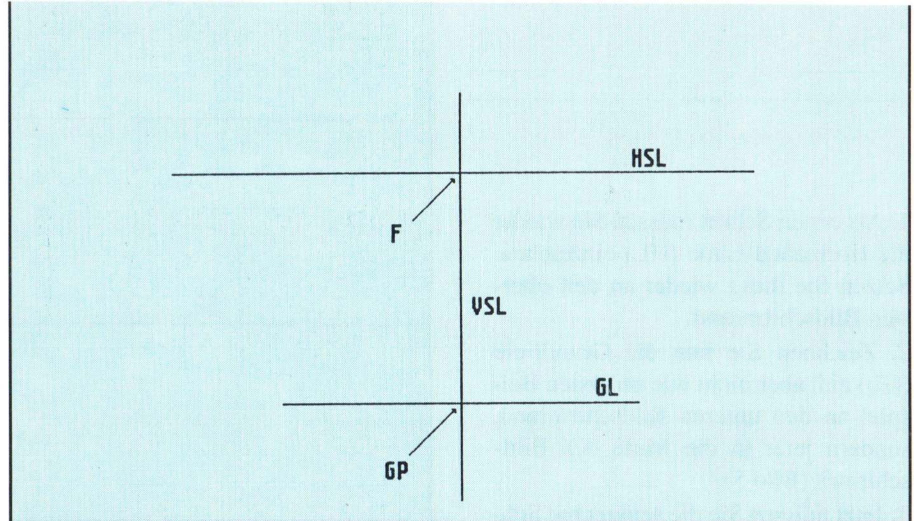
Immer dann, wenn Sie einen Raum oder eine Fläche dreidimensional darstellen möchten, brauchen Sie die perspektivische Frontansicht. Diese zu konstruieren ist leichter als es aussieht. Wenn Sie genau die vorgegebenen Schritte einhalten, dann können Sie Ihr Ziel eigentlich nicht verfehlen. Hier wird die Konstruktion anhand eines gekachelten Fußbodens gezeigt, wobei Sie dieses auch ohne weiteres für ein Schachbrett oder ähnliches benutzen können. Und mit etwas Phantasie ist es auch ohne große Mühe möglich, ein Zimmer mit Möbeln oder eine Landschaften mit Bergen, Fluß und Bäumen dreidimensional darzustellen.

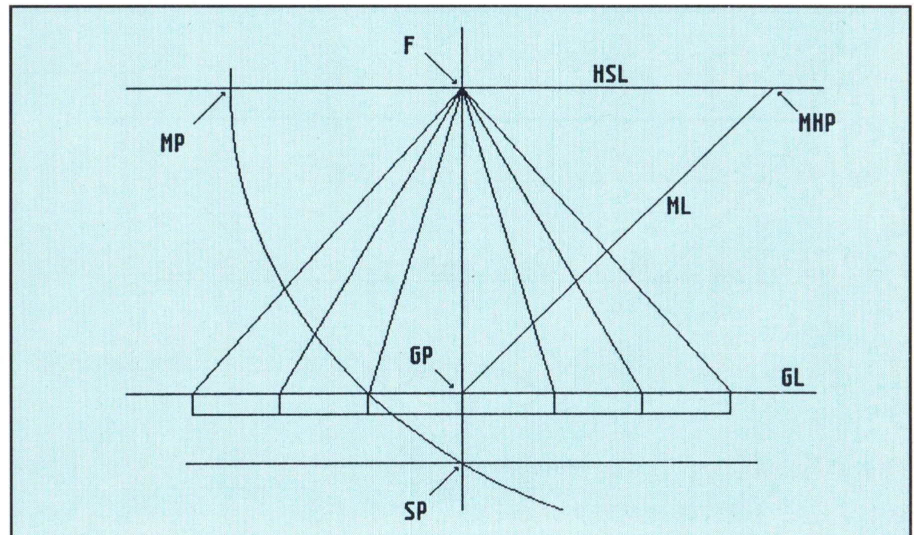
1. Zeichnen Sie im ersten Schritt die vertikale Sehlinie (VSL), auf der dann der Betrachter steht. Das Beste ist es, wenn Sie die vertikale Sehlinie (VSL) in der Mitte des Bildschirms plazieren.
2. Im nächsten Schritt zeichnen Sie die horizontale Sehlinie (HSL) ein. Diese sollte in der Augenhöhe des Betrachters liegen, also auf Ihrem Monitor etwa im obersten Teil des Bildschirmes. So ist auch die beste Platzverteilung, da unter HSL später noch das zu konstruierende Feld passen muß. An der Stelle, an der sich die vertikale Sehlinie (VSL) mit der horizontalen Sehlinie (HSL) schneidet, ist der Fluchtpunkt (F) für die Seitenlinien der Fläche (Bild 1).
3. Im dritten Schritt müssen Sie die Grundlinie (GL) für die zu konstruierende Fläche festlegen. Den Punkt, an dem sich die vertikale Sehlinie (VSL) und die Grundlinie (GL) schneiden, nennen wir Grundpunkt (GP).
4. Als nächstes müssen Sie nun die Breite der einzelnen Felder angeben, die sich auf der Fläche befinden. Zeichnen Sie Ihre Maße auf die Grundlinie (GL). Die vertikale Sehlinie (VSL) wird hier als Mittelpunkt für die anderen Verbindungslinien von der Grundlinie (GL) hin zum Fluchtpunkt (F) verwendet (Bild 2).
5. Jetzt müssen Sie den Standpunkt des Betrachters einzeichen, der sich auch wieder auf der vertikalen Sehlinie (VSL) befindet. Das Beste ist, wenn Sie diesen Punkt an den untersten Rand des Bildschirms setzen, denn später wird nichts mehr unter diesen Punkt gesetzt. Nehmen Sie nun die Entfernung von der horizontalen Sehlinie (HSL) bis auf die Höhe des Standpunktes und setzen Sie diese dann um 90 Grad gedreht auf die horizontale Sehlinie (HSL), mit dem linken Ende auf den Fluchtpunkt. An dem rechten Ende erhalten Sie den Mess-Hilfs-Punkt (MHP). Aktivieren Sie nun die Kreisfunktion Ihres Zeichenprogrammes, und zeichnen Sie einen Kreis um den Mess-Hilfs-Punkt (MHP), so daß der Kreis den Standpunkt schneidet. An dem Schnittpunkt des Kreises mit dem Mess-Hilfs-Punkt (MHP) mit der horizontalen Sehlinie (HSL) ist der Messpunkt (MP).
6. Zur Konstruktion der vertikalen Linien der Fläche müssen Sie jetzt nur noch den Grundpunkt (GP) mit dem Mess-Hilfs-Punkt (MHP) verbinden.

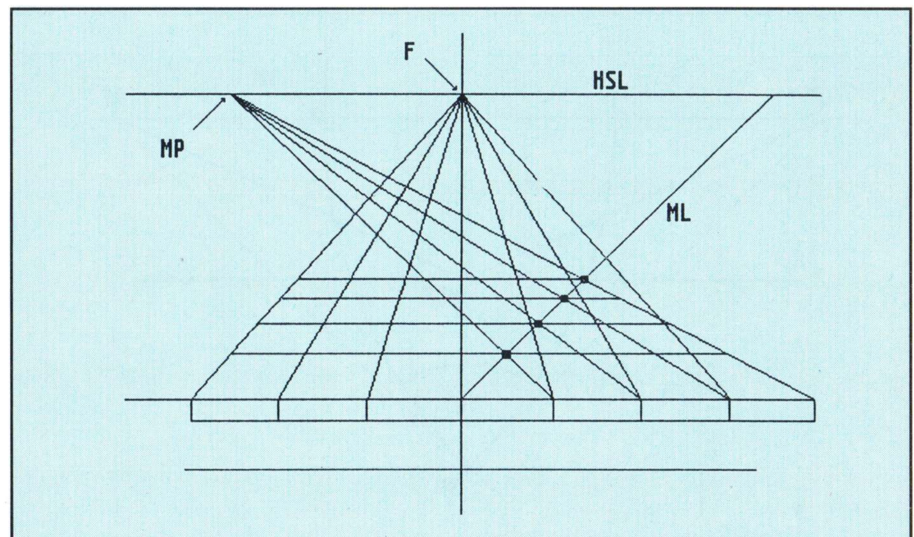
Jetzt noch Linien von dem Mess-Punkt (MP) zu den einzelnen Punkten auf Ihrer Grundlinie (GL) zeichnen (Bild 3), die die Breite der einzelnen Flächenteile im 'Bereich der Grundlinie (GL) angeben, und an den Stellen, an denen sich die Meßpunkt-Grundlinien-Geraden mit der Meß-Hilfspunkt-Grundlinien-Geraden schneiden, noch Linien parallel zur Grundlinie (GL) zeichnen, und die Fläche ist fertig konstruiert.

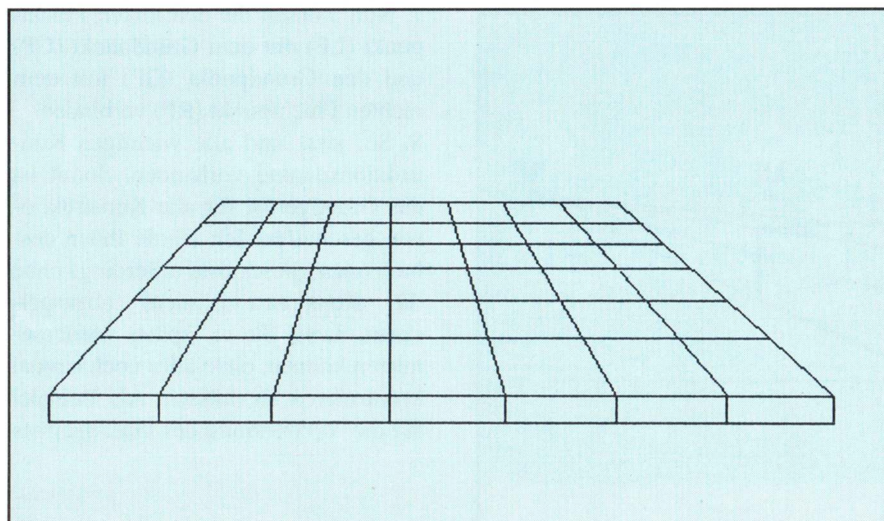
Na, das war doch gar nicht so kompliziert, wenn Sie sich genau an die Zeichnungen und den Text gehalten haben. Aber Sie können mit dem gleichen System auch eine schräge Perspektive erzeugen. Das Feld auf wird haargenau die gleiche Art konstruiert wie die Frontansicht, nur müssen Sie hier etwas genauer auf die horizontalen Linien achten, können das jedoch mit einem weiteren Fluchtpunkt zur Konstruktion der horizontalen Linien erleichtern, da diese in einer Schrägansicht nicht mehr parallel zueinander liegen. Wenn Ihr Standpunkt (SP) in der linken Ecke des Bildschirms liegt, dann muß der neue Fluchtpunkt auch links liegen. Das gleiche geht auch auf der rechten Seite (Bild 4).

Wenn Sie es nun nicht bei der Fläche belassen wollen, sondern vielleicht einen Raum mit zwei Wänden daraus machen wollen, dann ist das auch kein großer Aufwand mehr. Sie müssen jetzt das ganze Verfahren nur noch um 90 Grad gedreht durchführen und die rechte bzw. die linke Seite Ihrer zweiten Fläche an die rechte bzw. die linke Seite ihrer ersten Fläche anschließen. Wenn Sie mit Deluxe Paint II arbeiten, dann brauchen Sie nur die die Fläche als Brush zu nehmen und diesen dann um 90 Grad zu drehen, und schon haben Sie eine räumliche Röhre.









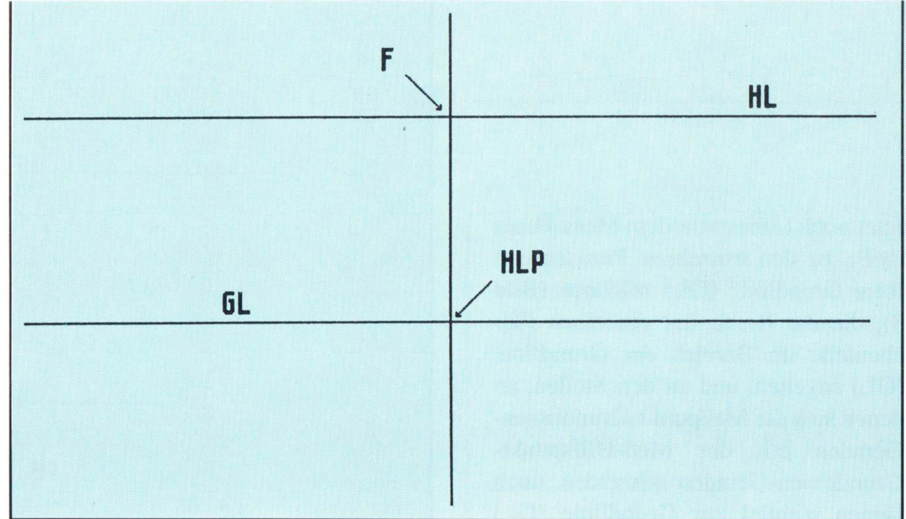
Die Konstruktion der perspektivischen Schrägansicht

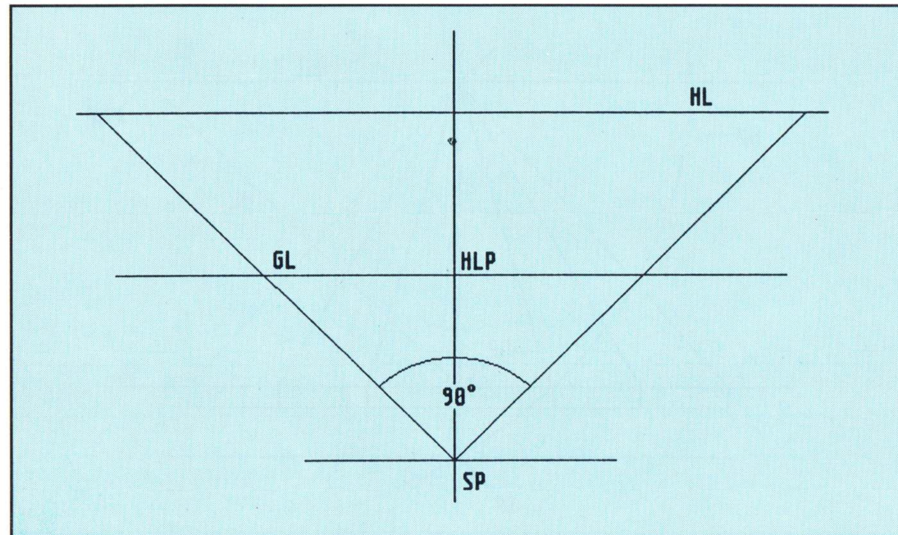
Diese Konstruktion, mit deren Hilfe Würfel oder ähnliche Gegenstände, die im Raum stehen, erstellt werden können, unterscheidet sich nicht sehr von der perspektivischen Frontansicht. Gehen Sie auch hier wieder streng nach der Reihenfolge, da Sie Sich sonst selbst aus dem Konzept bringen.

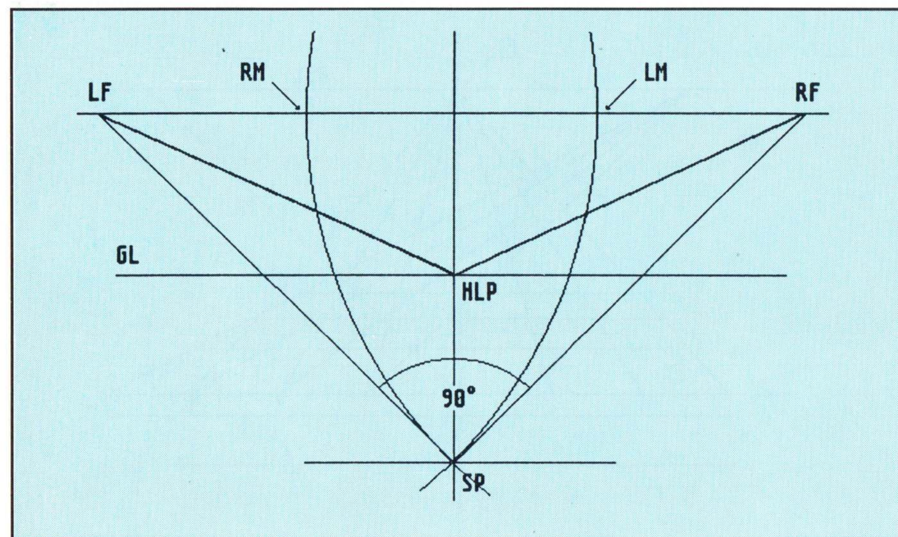
1. Als ersten Schritt müssen Sie wieder die Horizontal-Linie (HL) einzeichnen. Setzen Sie diese wieder an den obersten Bildschirmrand.

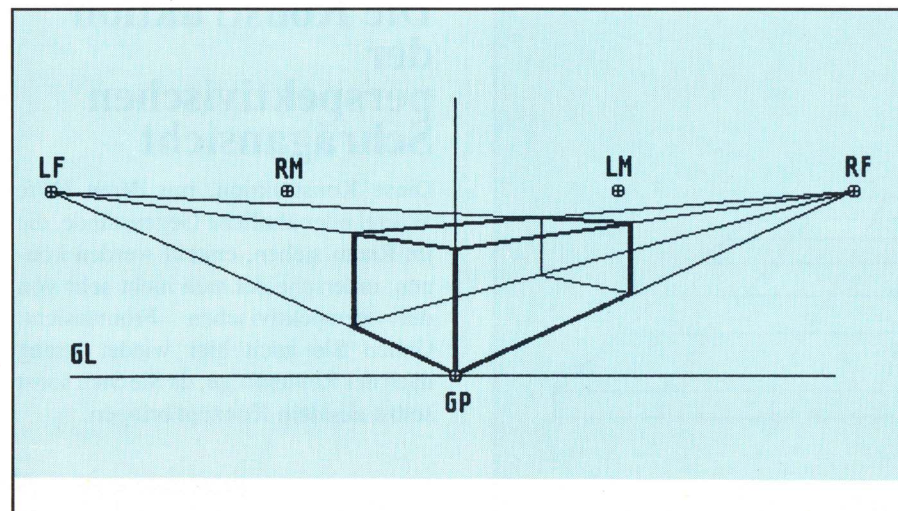
2. Zeichnen Sie nun die Grundlinie (GL) ein, aber nicht wie im ersten Beispiel an den unteren Bildschirmrand, sondern jetzt in die Mitte des Bildschirmes (Bild 5).

3. Jetzt müssen Sie die senkrechte Sehlinie (SL) etwa in die Mitte des Bildschirmes einzeichnen; den Punkt, an dem sich die senkrechte Sehlinie mit der Horizontallinie schneidet, nennen wir Horizontalpunkt (HLP), den Punkt, an dem sich die Grundlinie (GL) mit der senkrechten Sehlinie (SL) schneidet, nennen wir Grundpunkt (GP).









4. Im vierten Schritt zeichnen Sie nun auf die senkrechte Sehlinie (SL) im unteren Bildschirmbereich den Standpunkt (SP) ein (Bild 6).

5. Zeichnen Sie jetzt vom Standpunkt (SP) aus einen Winkel von 90 Grad, wobei die Winkelhalbierende die senkrechte Sehlinie (SL) ist. Verlängern Sie nun die Winkelschenkel bis auf die Horizontallinie (HL). Dort tragen Sie beim linken Schnittpunkt den linken Fluchtpunkt (LF) und beim rechten Schnittpunkt den rechten Fluchtpunkt (RF) ein (Bild 7).

6. Nun müssen Sie den rechten (RM) und den linken (LM) Meßpunkt einzeichnen. Dazu zeichnen Sie einen Kreis um den rechten (RF) bzw. den linken (LF) Fluchtpunkt mit dem Radius der Entfernung des jeweiligen Fluchtpunktes zum Standpunkt (SP). Beim Kreis um den rechten Fluchtpunkt entsteht auf der Schnittstelle mit der Horizantallinie (HL) der linke Mittelpunkt (LM), beim Kreis um den linken Fluchtpunkt der rechte Mittelpunkt (RM).

7. Nun müssen Sie den linken Fluchtpunkt (LF) mit dem Grundpunkt (GP) und den Grundpunkt (GP) mit dem rechten Fluchtpunkt (RF) verbinden.

8. So, jetzt sind alle wichtigen Konstruktionspunkte vorhanden. Somit ist das Grundgerüst für alle Konstruktionen geschaffen. Ich würde Ihnen deshalb raten, dieses Bild (allerdings ohne die Konstruktionslinien) abzuspeichern, damit Sie es später wiederbenutzen können, ohne alles noch einmal konstruieren zu müssen. Als Beispiel für die Verwendung des Grundgerüsts

werden wir nun einen Würfel konstruieren.

1. Zeichnen Sie sich von dem Grundpunkt (GP) aus eine Garde nach oben, die die Höhe des Würfels festlegt. Den oberen Punkt nennen wir A. Sie müssen nun den Grundpunkt (GP) jeweils mit dem rechten (RF) und dem linken (LF) Fluchtpunkt verbinden. Tragen Sie nun die rechte und die linke untere Ecke des Würfels an. Mit diesen drei Linien sind die Größenverhältnisse des Würfels festgelegt. Alle anderen Eckpunkte werden nun konstruiert.

2. Für die oberen Kanten müssen Sie den Punkt A mit dem rechten (RM) und dem linken (LM) Mittelpunkt verbinden. Jetzt wird von der linken bzw. der rechten unteren Ecke eine Senkrechte nach oben gezeichnet, bis sie die Verbindungslinie des Punktes A mit dem rechten Fluchtpunkt (RF) schneidet bzw. bis sie die Verbindungslinie des Punktes A mit dem linken Fluchtpunkt (LF) schneidet. Dadurch entstehen zwei Punkte; den linken nennen wir B und den rechten nennen wir C. Um alle Kanten des Würfels vollständig antragen zu können, müssen Sie nur noch den Punkt B mit Ihrem rechten Fluchtpunkt (RF) und den Punkt C mit dem linken Fluchtpunkt (LF) verbinden. Wenn Sie aber jetzt einen Würfel aus Glas darstellen möchten, dann müssen die verdeckten Kanten noch eingezeichnet werden. Verbinden Sie dazu den linken Fluchtpunkt (LF) mit der rechten unteren Ecke Ihres Würfels und den rechten Fluchtpunkt (RF) mit der linken unteren Ecke, und schon haben Sie einen Glaswürfel.

Das war's für diese Ausgabe, und noch viel Spaß beim Nachzeichnen!

ENDE

WANTED!

PD-SOFTWARE GESUCHT!

Belohnung: 5 PD-Disketten

Wer aufmerksam das PD-Angebot studiert, wird schnell feststellen, daß fast alle Programme und auch die meisten Grafik- und Sounddemos von amerikanischen und englischen Programmierern erstellt werden. Dies hat unter anderem zur Folge, daß fast alle Kommentare in den Programmen und vor allem die Anleitungen in englisch verfaßt sind. Dies macht es manchem deutschen AMIGA-Einsteiger nicht leicht, den AMIGA und dessen Programmierung kennenzulernen. Wir wollen deshalb noch einmal alle Programmierer im deutschsprachigen Raum aufrufen, ihre selbstgeschriebenen Programme, die sonst in einer dunklen Schublade verschwinden, in die PD-Sammlung zu geben.

Als kleinen Anreiz erhält jeder Programmierer, dessen Programm in den KICKSTART PD SERVICE aufgenommen wird, fünf PD-Disketten seiner Wahl.

C. BERGMEIER

DIE SPIELE-SCHMIEDE

Zu Gast bei Rainbow Arts

RAINBOW ARTS ist eines der erfolgreichsten deutschen Softwarehäuser. Mit Titeln wie THE WALL, THUNDERBOY, JINKS, GARRISON, GARRISON II und vor allem mit THE GREAT GIANA SISTERS konnte das junge Team aus Gütersloh die Spielefans begeistern. Thomas Hertzler (TH), seines Zeichens Entwicklungsleiter bei Rainbow Arts, überwacht die Qualität aller Produkte, koordiniert die Entwicklungsphasen und betreut die zahlreichen Programmierer bei ihrer Arbeit. Das vorliegende Interview mit Thomas und Rainbow Arts-Geschäftsführer Marc Ullrich (MC) entstand bei einem Besuch unseres Mitarbeiters Carsten Borgmeier in der "Gütersloher Spieleschmiede".

CB:
Thomas, wie bist Du Entwicklungsleiter bei Rainbow Arts geworden?

TH:
Vor etwa zwei Jahren, ich steckte gerade in meinem Elektrostudium, entdeckte ich eine Anzeige in einem Computermagazin. Rainbow Arts, damals noch Herstellerfirma von Anwendersoftware, suchte Programmierer für die Entwicklung von Computerspielen. An meinem Studium hatte ich keinen Spaß, da mir das Theoretische nicht liegt. Also beschloß ich, mich auf die Anzeige zu bewerben. Dazu programmierte ich mein erstes Computerspiel, was auch gleichzeitig mein erstes größeres Assemblerprojekt war, und schickte es an Rainbow Arts. Das Spiel wurde unter dem Titel Skyfighter für Atari ST und Amiga veröffentlicht. Ich möchte mich an dieser Stelle für das meiner Meinung nach mißlungene Spiel entschuldigen. Skyfighter konnte kein gutes Spiel werden, da die Gebiete Grafik, Sound und Programmierung nicht unter Spezialisten aufgeteilt waren. Ich habe das Spiel ganz alleine geschrieben. Heutzutage braucht man für jedes Gebiet Spezialisten, um erstklassige Spiele herstellen zu können. Nachdem ich mich mit meinem "Skyfighter" beworben hatte, wurde ich festangestellter Programmierer bei Rainbow Arts.
Bad Cat und die Konvertierungen der C64-Version von The Great Giana Sisters auf Atari ST und Amiga waren meine nächsten Projekte. Damals war Marc noch Geschäftsführer und eine Art Entwicklungsleiter in einer Person. Da ihn die Geschäftsführung voll beanspruchte, hatte er nicht soviel Zeit, sich um die Organisation der Entwicklungsabteilung zu kümmern, so daß ich dazu herangezogen wurde.

CB:
Hast Du Dir in Punkto Organisationstechniken noch Kenntnisse aneignen müssen, oder warst Du für diese neue Aufgabe schon qualifiziert?

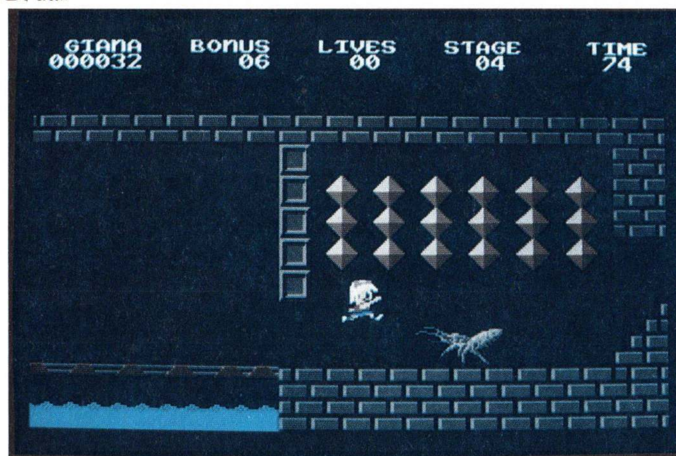
TH:
Weder, noch. Da bin ich hereingewachsen. Bei einem etablierten Softwarehaus wie beispielsweise Ariolasoft hätte ich sicherlich Referenzen vorweisen müssen. Da Rainbow Arts sich aber erst allmählich entwickelt hat, brauchte ich das nicht. Ich bemühe mich, die Entwicklunsabläufe nach meinen eigenen, laienhaften Vorstel-

lungen zu organisieren. Dabei versuche ich, die Entwicklung zu straffen, so daß nicht alles aneinander vorbeiläuft. Jedes Mitglied im Entwicklungsteam eines Spiels muß wissen, was der andere macht. Der Programmierer muß sich mit dem Grafiker absprechen, der Soundspezialist muß vom Programmierer wissen, welche Soundeffekte er an welchen Stellen im Spiel programmieren soll. Als es bei Rainbow Arts noch keinen Entwicklungsleiter gab, haben wir jede Woche eine Konferenz abgehalten. In der sogenannten W.O.P. (Wöchentliche Organisations- und Problemkonferenz) wurden Konzepte besprochen und Grafiker, Programmierer und Soundspezialist aufeinander abgestimmt. Diese Verfahrensweise war nicht effektiv, da wir bei Rainbow Arts immer gleichzeitig an mehreren Spielen arbeiten. Wenn sich das Entwicklungsteam A über Spiel A unterhielt, hatte Entwicklungsteam B, das an Spiel B arbeitet, Leerlauf und langweilte sich solange, bis man über ihr Projekt sprach. Die ganze Arbeitsleistung des Betriebes war bei diesen Konferenzen für mehrere Stunden blockiert. Um das Ganze effizienter zu gestalten, spricht jetzt der Entwicklungsleiter mit jedem Entwicklungsteam deren Belange durch. Alle anderen Mitarbeiter, die nichts mit dem Projekt zu tun haben, werden so nicht von ihrer Arbeit abgehalten. Das einzige, was jetzt noch im großen Kreis besprochen wird, sind die Spielkonzepte, weil es da unverzichtbar ist, daß mehrere Leute ihren kreativen Senf auf einen Haufen werfen.

CB:
Wie sieht Dein typischer Arbeitstag aus?

TH:
Mein Arbeitstag beginnt um neun Uhr morgens. Zuerst gehe ich die Post durch. Ich sehe mir alle Bewerbungen an. Wenn Leute uns ihr fertiges Programm zum Vertrieb anbieten, prüfe ich die Qualität. Falls bei den Bewerbungen Demodisketten oder Videos beiliegen, schaue ich mir auch diese genau an. Außerdem sehe ich Bewerbungen von Programmierern durch, die als freischaffende oder festangestellte Programmierer für uns arbeiten wollen. Meistens sind mit diesen Bewerbungen Telefongespräche gekoppelt. Nachmittags rufe ich die zumeist schulpflichtigen Jugendlichen an und spreche mit ihnen über eine mögliche Zusammenarbeit oder sage ab, und arbeite an den Spieleprojekten, in die ich involviert bin. Meine letzten Projekte waren Great Giana Sisters auf Atari ST und Amiga. Außerdem entwickle ich noch firmeninterne Programmiertools, koordiniere die Arbeit der festangestellten Programmierer und betreue alle freien Mitarbeiter.

CB:
Hast Du auch eigentlich auch einen Amiga zu Hause stehen, oder wirst Du dort dem Amiga untreu?



The Great Giana Sisters ist der momentane Renner von Rainbow Arts.

TH:
Nein, zu Hause habe ich überhaupt keinen Computer. Wenn ich nach einem zwölfstündigen Arbeitstag nach Hause komme, dann ist ein Computer das letzte, was ich noch sehen möchte.

CB:
Wieviele Programmierer arbeiten für Rainbow Arts?

TH:
Zur Zeit arbeiten 12 festangestellte und ca. 40 freie Mitarbeiter für uns.

CB:
Welche Voraussetzungen muß man mitbringen, um bei Rainbow Arts als festangestellter Programmierer arbeiten zu können? Muß man ein Informatikstudium oder eine EDV-Ausbildung nachweisen?

TH:
Nein, ich glaube, in dieser Beziehung ist Rainbow Arts Deutschlands unkonventionellste Firma. Unsere festangestellten Programmierer haben sich teilweise auf Postkarten beworben. Wenn ein Bewerber mindestens einen Homecomputer perfekt beherrscht und uns eine gute Demo abliefert, hat er große Chancen. Uns ist völlig egal, was man für eine Ausbildung genossen hat, Hauptsache, man kann gut programmieren. Man sollte flexibel sein und sich nicht auf einen Rechner versteifen. Leute, die ihren Computer über alles lieben und nicht auf anderen Systemen arbeiten wollen, haben keine Chance. In diesem Geschäft muß man flexibel sein, weil innerhalb weniger Jahre der Rechner, auf den man spezialisiert ist, durch einen anderen Computer vom Markt verdrängt werden kann. Wenn uns die Demos eines Bewerbers auf eine Festanstellung gefallen, laden wir ihn nach Gütersloh zu einem Vorstellungsgespräch ein. In diesem Gespräch wollen wir den Charakter eines Bewerbers prüfen. Wir versuchen, herauszufinden, ob er aus der Crackerszene kommt und wie loyal er ist. Außerdem ziehen wir Erkundigungen über ihn ein. Wir haben nämlich panische Angst davor, daß sich ein Raubkopierer aus der Szene bei uns einschleicht. Wenn Muster vor Markteinführung in den "Sumpf" gehen würden, wäre das fatal für uns. Wir müssen den Charakter des Bewerbers genau prüfen, um unliebsame Überraschungen zu vermeiden. Hinzu kommt noch, daß man sich in die Gemeinschaft hier einfügen muß.

CB:
Für welche Rechner produziert Ihr im Moment Eure Spiele?

TH:
Im Moment bieten wir Spiele für C64, Amstrad, Atari ST, Amiga und MS-DOS-Rechner an.

CB:
Welche Systeme könnten in der Zukunft interessant werden?

TH:
Es könnte die Atari 800-Reihe interessant werden, wenn sich das Telespiel noch weiter durchsetzt und man die Cassetten auch für die Ataricomputer benutzen kann.

CB:
Was ist mit dem Archimedes, dem neuen Wundercomputer?

TH:
Für den Archimedes lohnt sich die Spieleentwicklung noch nicht. Der prozentuale Anteil am Markt ist viel zu klein. Meiner Meinung nach entwickelt sich der Archimedes bei einem Kaufpreis für das kleinste Gerät von 3300 DM (ohne Monitor) eher zu einer Workstation und nicht zu einer Spielmaschine. Der Archimedes ist keine klassische Spielemaschine wie beispielsweise Amiga und C64, auch wenn die Fähigkeiten dazu vorhanden sind. Wenn der Anteil am Markt jedoch entsprechend größer wird, werden wir auch Spiele für den Archimedes produzieren.

CB:
Wie sieht die Entwicklung eines Computerspiels aus?

TH:
Zuerst benötigt man ein Konzept, das Vorstellungen festlegt, wie das Spiel nachher auszusehen hat und was der Spieler im fertigen Spiel anstellen kann. Der Programmierer bekommt das Konzept und die Grafiken. Nach vier Wochen Programmierzeit trudelt meist die erste Demo bei uns ein. Ein Männchen läuft über den Bildschirm, und der Screen scrollt. Vom Spiel an sich ist noch nicht viel zu erkennen. Der Programmierer eleminiert nach einer gründlichen Inspektion kleine Fehler wie zum Beispiel Ruckeln oder Flackern beim Scrolling. Der ganze Vorgang wiederholt sich solange, bis das Rohgerüst des Spiels steht. Wenn das der Fall ist, beginnt die zweite Entwicklungsphase, das sogenannte Feintuning. Diese Phase haben wir eingeführt, als wir merkten, daß die Fachpresse in der Punkteverteilung immer Abstriche bei der Spielbarkeit unserer Spiele machte. Beim Feintuning wird nur noch die Spielbarkeit überprüft. Kann man den Held schnell genug steuern? Sind irgendwo linke Stellen im Spiel, oder verliert man in einem Level alle 10 Bildschirmleben? Ist die Steuerung im Programm komfortabel genug? Alle Mängel werden behoben, bevor das Spiel zum Kopierwerk geht; denn unsere Entwicklungsphilosophie ist folgendermaßen: Computerspiele sollen nicht nur einen guten Sound und eine gute Grafik haben, die Games müssen auch Spaß machen und eine lange Motivation bieten. Zum ersten Mal haben wir das Feintuning bei The Great Giana Sisters eingesetzt. Der Erfolg mit diesem Spiel rechtfertigt unsere Feintuningphase.



Starballs erinnert vom Grundprinzip aus stark an die ersten Tennis-Telespiel-Varianten.

CB:
Wie teuer ist die Entwicklung eines Computerspiels für Rainbow Arts?

TH:
Das kann ich Dir nicht sagen, darüber habe ich keinen Überblick. Marc weiß da besser Bescheid.

MU:
Die Entwicklung eines Spiels für die Systeme Schneider, C64, Amiga und Atari ST kostet etwa 100000 Mark.

CB:
Wieviele Spiele hat Rainbow Arts bisher veröffentlicht?

MU:
Im Moment sind etwa 40 Spiele von uns auf dem Markt.

CB:
Wie würdet Ihr die Lage auf dem Programmierermarkt beurteilen? Gibt es viele fähige Spieleprogrammierer für dem Amiga?

TH:
Es gibt viele technisch ausgereifte Programmierer. Leider können nur sehr wenige auch gute Computerspiele entwickeln. Es reicht eben nicht, eine tolle Grafik ablaufen zu lassen und einen bombastischen Sound zu spielen. Die Programmierer müssen ein Spiel programmieren können, das Spaß macht. Dieses Feeling haben leider nur sehr wenige.

MU:
Ein großer Fehler der freischaffenden Amigaprogrammierer liegt darin, daß sie nur abgucken, und das auch noch schlecht. Mittlerweile wurden uns 50 PacMan-Varianten angeboten, die mit Sicherheit nicht veröffentlicht werden. In Shoot'em Ups können wir auch schon schwimmen. Jeder Clone gleicht wie ein Ei dem anderen. Die meisten Leute kupfern eine Idee eines bekannten C64-Spiels schlecht ab, schicken es uns zu und wollen, daß wir die 7698. PacMan-Variante veröffentlichen.

CB:
Was für Computerspiele sollen Euch denn die freischaffenden Programmierer für Spiele schikken?

MU:
Die Spiele sollen ein innovatives Konzept haben, langen Spielspaß und eine hohe Motivation garantieren. Ich kann keine PacMan, Shoot'em ups und Tron-Verschnitte mehr sehen. Die wandern sofort in die Absagekiste.

CB:
Habt Ihr Lieblingsspiele?

TH:
Mir fällt im Moment keines ein. Ich spiele nicht so viel.

MU:
Na klar: Great Giana Sisters, Jinks, Garrison ... (Gelächter). Nein, Scherz beiseite. Ich hasse es, unsere eigenen Produkte zu nennen. Ich zähle mal ein paar Konkurrenzspiele auf: Marble Madness finde ich klasse. Bard's Tale, Hitchhiker's Guide to the Galaxy, Jumpman und Skramble haben mir auch sehr gefallen.

CB:
Hat Rainbow Arts schon einmal Probleme mit Viren gehabt?

TH:
Ja, leider! Die Leute, die diese Viren in Umlauf gebracht haben, sollte man hinter Gitter bringen. In meinen Augen sollte man diese Freaks stärker als Raubkopierer bestrafen. Bei uns war der ganze Betrieb einmal für mehrere Stunden lahmgelegt. Ein Programmierer entdeckte, daß sein Spiel nach dem Starten nicht richtig lief. Stundenlang suchte er den Fehler. Seine Kollegen unterstützten ihn bei der Fehlersuche, bis einer vorschlug, die Spieldiskette auf Viren zu überprüfen. Viren sind für ein Softwarehaus das schlimmste, was es gibt. Wenn unsere kreative Arbeit, an der wir monatelang gearbeitet haben, durch Viren zerstört wird, entstehen astronomische Schäden. Alle Arbeit war umsonst. Meiner Meinung nach kann man diese "Virenspreader" gar nicht hart genug bestrafen.

CB:
Zum Abschluß möchte ich noch wissen, welche neuen Spiele Rainbow Arts bald auf den Markt bringt.

TH:
Wir sitzen gerade an der Programmierung von GIANA SISTERS II. Das Spiel ist komplett anders als der Vorgänger. Geblieben ist, daß man mit Giana von links nach rechts läuft. Alles andere ist komplett neu. Es gibt verschiedene Spielmodi, neue Gegner und Hindernisse. Mehr möchte ich dazu aber noch nicht verraten, da wir gerade erst mit dem Projekt begonnen haben. Ein weiteres Spiel nennt sich CHARMING ORX. Der Spieler steuert ein knuddeliges Monster und muß sich mit Hilfe seines Jojos durch diverse Level kämpfen und dabei andere Monster unschädlich machen. STARBALLS ist eine Pong-Variante, die man zu zweit spielen kann. Jeder Spieler steuert einen Schläger und muß versuchen, eine Kugel so auf die andere Seite des Bildschirms zu schießen, daß der Gegner sie nicht erreichen kann. Das Spiel zeichnet sich durch exellente Hintergrundgrafiken und durch verschiedene Spielmodi aus. Eventuell bauen wir eine Modemoption ein. Für HOLLYWOOD CLASSICS steht das Konzept noch nicht ganz. Nur so viel ist sicher: Der Spieler kann fünf klassische Hollywoodfilme nachspielen. Darunter werden "Denn sie wissen nicht, was sie tun" mit James Dean und "Star Wars" sein. Über die anderen Filme sind wir uns noch nicht einig. In GRAFFITY MAN läuft der Held durch verschiedene Level und muß am Ende eines jeden Abschnitts ein Graffity mit einer Spraydose nachmalen. Das sind im Moment alle neuen Amiga-Projekte.

Carsten Borgmeier:
Ich danke Euch für das interessante Gespräch.

ENDE



AMIGA 500/1000
DAS EINSTEIGERBUCH
Thomas Binzinger

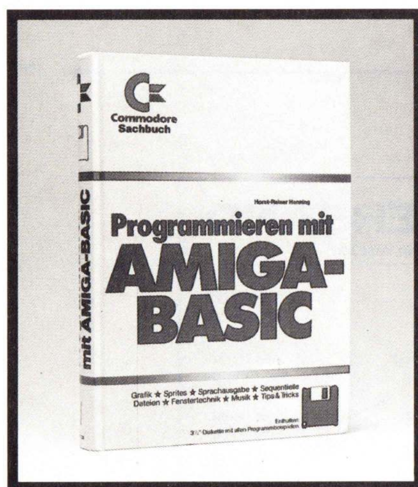
280 Seiten
SYBEX-Verlag GmbH
Preis: 29.80 DM

SYBEX vertreibt in seiner umfangreichen Angebotspalette unter anderem ein Buch, das für den Amiga-Einsteiger gedacht sein soll. Das 280 Seiten starke Buch verspricht zumindest von den in der Inhaltsangabe angedeuteten Kapiteln, ein nützlicher Wegbereiter für den Amiga-Neuling zu sein. Insgesamt findet der Käufer 12 verschiedene Kapitel, angefangen von "Die ersten Schritte" über "Der Kommandointerpreter CLI" bis hin zu "Tips und Tricks". Des weiteren sind die einzelnen Themen mit Illustrationen versehen, so daß einige, für den Anfänger schwer verständliche Bereiche leichter zu überschauen sind.
Der Autor Thomas Binzinger hat sein Werk in einen lockeren Schreibstil, ohne schwierige Fremdwörter oder komplexe und verwirrende Erklärungen, verfaßt. Dadurch eignet sich das Buch als Lektüre für den absoluten Amiga-Anfänger. Wer bereits mit einem anderen Computer arbeitet oder gearbeitet hat, der kann einige (Teil-) Kapitel getrost überspringen, da auf grundlegende Dinge, die nicht als Amiga-spezifisch bezeichnet werden können, eingegangen wird. Obwohl der Buchtitel suggeriert, daß sich das Werk lediglich mit den Amiga-Typen 500 und 1000 befaßt, kann auch ein Amiga 2000-Neuling die Lektüre zur Hand nehmen. Spezielle Amiga 2000-Kapitel sind aber nicht zu finden.

Fazit:

"Amiga 500/1000 - Das Einsteigerbuch" ist nicht als Nachschlagewerk geeignet, sondern kann als abendliche Bettlektüre herangezogen werden. Die einzelnen Themen werden gut verständlich für den Amiga-Neuling abgehandelt und geben einen generellen Überblick. Tiefergehende Informationen findet der Leser hingegen kaum, so daß bei Bedarf auf anderweitige Literatur zurückgegriffen werden muß. Wer mit dem Amiga schon ein wenig vertraut ist, der kann auf das Buch verzichten. Für absolute Amiga- und Computer-Neulinge kann es aber ein hilfreiches Werk darstellen.

(AK)

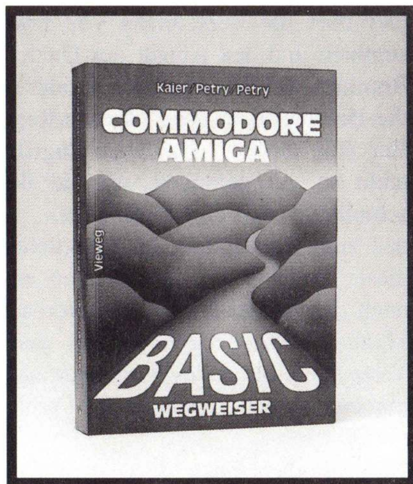


PROGRAMMIEREN MIT AMIGA-BASIC
Horst-Rainer Henning

370 Seiten
Markt&Technik Verlag AG
Preis: 59.-

BASIC ist für den AMIGA aus mindestens zwei Gründen interessant: zum einen ist ein Interpreter im Lieferumfang enthalten, zum anderen ist die Sprache gerade für Anfänger wie geschaffen. Das Haus Markt&Technik vertreibt als Unterstützung dieser Programmiersprache ein ca. 370 Seiten starkes Buch. Speziell greift das Buch ausgewählte Themen des Amiga auf, soll demnach kein Nachschlagewerk für jeden in Amiga-BASIC zur Verfügung stehenden Befehl darstellen, sondern geht vielmehr auf die Programmierung der Amiga-spezifischen Eigenschaften ein.
Grundsätzlich ist es in sieben Kapitel unterteilt, angefangen mit "Grundlagen von Amiga-BASIC", das die allgemeine Arbeitsweise mit dem Interpreter beschreibt und des weiteren auf gängige BASIC-Befehle eingeht. Die verschiedenen Befehle werden außerdem in mehr oder weniger großen Programmbeispielen näher erläutert. Das zweite Kapitel ist der Grafik des Amiga gewidmet. Befehle der Grafikoberfläche sowie einzelne Grafikbefehle werden hier abgehandelt.
Bevor Sprites, Bobs und Sound behandelt werden, ist der Dateiprogrammierung ein Kapitel gewidmet. Weiter geht mit Tips und Tricks, einer globale BASIC-Befehlsübersicht und natürlich einem Anhang.
Alle Kapitel sind mit Beispielsprogrammen versehen, die abgetippt werden können oder auf der mitgelieferten Programmdiskette zu finden sind.
"Programmieren mit AMIGA-BASIC" ist sowohl für den Anfänger als auch für den fortgeschrittenen Programmierer gedacht. Reine Anfänger werden wohl manchmal Schwierigkeiten bekommen, denn obwohl auch viel Grundlegendes erklärt wird, wird auf die allgemeine Programmierung von BASIC nicht detailliert eingegangen. Fortgeschrittene BASIC-Programmierer werden hingegen unterfordert, obwohl sich manch nützlicher Tip findet. Aus diesem Grund kann das Buch mit ruhigen Gewissen nur den Leuten empfohlen werden, die schon ein wenig BASIC beherschen, sich aber noch lange nicht zu den Profis zählen können.

(AK)



COMMODORE AMIGA BASIC-WEGWEISER
KAIER / PETRY / PETRY

500 Seiten
Vieweg-Verlag
Preis: 76.- DM

Drei Autoren zeichnen sich für das hier beschriebene Werk verantwortlich. Das fast 500 Seiten dicke Buch ist nicht nur ein BASIC-Buch, sondern greift zusätzlich im ersten Kapitel die globale Informatik auf. Das zweite Kapitel beschäftigt sich mit der Bedienung des Amiga und des Amiga-BASIC. Hier werden beipielsweise die nützlichsten CLI-Befehle näher erläutert oder Stapel-Dateien beschrieben. Wie in allen Kapiteln sind auch hier Beispiele zum Eingeben und Illustrationen beigefügt. Leider wirken die Skizzen und Tabellen des öfteren leicht unübersichtlich, zu etwas mehr Sorgfalt wäre zu raten gewesen.

Das dritte Kapitel beschäftigt sich dann mit der eigentlichen Programmierung des BASIC. Dabei wird auch Wert auf Stil und Programmgliederung Wert gelegt.

Im Grunde genommen ist das Buch nicht nur für den Amiga-Anwender geschrieben, sondern beinhaltet zum größten Teil Gebiete, die auch auf andere Rechner übertragbar sind. Einige Abschnitte greifen jedoch Amiga-spezifisches auf.

Der Stil der Autoren ist trocken, ohne viele Schnörkel und Umschweife erhält der Leser die Information.

Die abgedruckten Programmbeispiele kann der Leser für den üppigen Preis von 48.- DM auf einer Diskette nachbestellen, was aber nicht umbedingt von nöten ist, da die Listings meistens recht kurz gehalten sind.

Fazit:

Der "BASIC-Wegweiser" für den Commodore Amiga entpuppt sich wahrlich als Wegweiser für einen BASIC-Programmierer. Der Leser erhält kein reines Nachschlagewerk für das Amiga-BASIC, sondern vielmehr eine ersthafte Lektüre für den, der BASIC von grundauf erlernen möchte. Die Amiga-spezifischen Befehle werden in einem Extrakapitel näher beschrieben. Dies erweckt den Anschein, daß große Teile des Buches von ähnlich lautenden Büchern für andere Rechner einfach übernommen wurden. Um einmal schnell eine Befehlssyntax nachzuschlagen, ist das Buch nicht geeignet. Anwendern, die des BASIC mächtig sind, kann vom Kauf dieses Buches abgeraten werden. Zu viele Kapitel sind lediglich für den Einsteiger bestimmt.

(AK)



C FÜR EINSTEIGER
Dirk Schaun

300 Seiten, 2. Auflage
Data Becker GmbH
Preis: 39.- DM

Data Becker wartet mit einer ganzen Reihe von Büchern für den Amiga auf. Eines davon ist "C für Einsteiger" von Dirk Schaun. Uns lag die zweite und erweiterte, knapp 300 Seiten starke Auflage vor.

Der Autor hat sein Werk, wie allgemein üblich, in Kapitel unterteilt, die mehr oder weniger aufeinander aufbauen. Der Schreibstil ist einfach und leicht verständlich. Der Autor sah von der leider weit verbreiteten "Fremdwörterprotzerei" ab.

Das Werk kann grob in zwei Teile gegliedert werden. Zunächst beschreibt der Autor grundlegende Dinge, bevor er ausführlich auf die Programmierung eingeht. Die Erklärungen bauen zum Teil aufeinander auf und gehen immer tiefer auf die C-Materie ein.

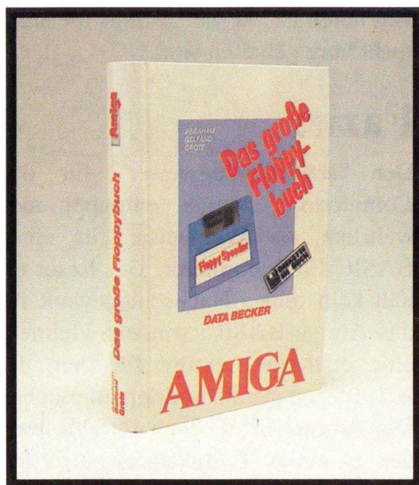
Ein besonderes Kapitel ist speziell der Amiga-Programmierung gewidmet. Hier findet der Leser unter anderen Programmbeispiele, die sich mit der Intuition oder der Grafikausgabe befassen. Ansonsten ist auf größere Programme verzichtet worden. Natürlich kann das Kapitel nicht die gesamten Amiga-Funktionen abdecken; hierfür sind eigenständige Bücher erhältlich.

Fazit

C-Einsteigern (und was liegt beim Amiga näher, als C zu lernen, will man diese Maschine effektiv programmieren) kann das Buch durchaus empfohlen werden. Kapitel wie "C an einem Wochenende" sollten meiner Meinung nach allerdings umbenannt oder weggelassen werden, denn dem Leser wir dadurch nur suggeriert, er könnte diese komplexe Sprache in dieser knappen Zeit erlernen.

Der Leser erhält ein Buch, das dem Neuling die Programmierung in C von Grund auf näher bringt. Manche Themen sind nach meinem Verständnis etwas kurz abgehandelt worden, so daß bei dem einen oder anderen Leser Verständnisprobleme auftreten können.

(AK)



DAS GROSSE FLOPPYBUCH
Abraham / Gelfand / Grote

400 Seiten
Data Becker GmbH
Preis: DM 59.-

"Die Floppy - unentbehrliches Hilfsmittel für die tägliche Arbeit mit dem Amiga."

So fängt das Vorwort der neuesten Data Becker-Publikation an. Der Düsseldorfer Verlag wird mit diesem Buch wieder einmal seinem Ruf gerecht und bringt als erster ein Buch zum Thema Disketten auf den Markt.

Das Buch ist mit 400 Seiten und der mitgelieferten Diskette zu einem stattlichen Schmöcker angewachsen. Dem Leser wird damit ein breiter Überblick über die Funktionen und Fähigkeiten der Amiga-Diskettenverwaltung gegeben. Um hier nicht eine chaotische (das Wort Chaot hatte vor Zimmermann noch eine andere Bedeutung) Anhäufung von Informationen abzuliefern, haben die Autoren, ihren Fachbereichen entsprechend, das Buch in drei Teile gegliedert.

Der erste Teil beschäftigt sich mit den Grundfunktionen der Workbench und des CLI, soweit sie die Diskettenoperationen betreffen. Der Neuling und Systemumsteiger erhält hier einen Einblick in die diskettenspezischen AmigaDos-Befehle, wie z.B. Assign, oder den Disketten-Arzt DiskDoctor. Im Anschluß an diesen Teil zeigen einige Beispiele den Diskettenumgang von Basic aus. Leider wird hier nicht auf die direkte Programmierung des Trackdiskdevice eingegangen, was sicher auch für Basicprogrammierer eine interessante Sache ist. So sind die Beispiele des ersten Teils eher als Ergänzung zum Basichandbuch zu sehen.

Der zweite Teil das Floppybuchs beschäftigt sich mit der Schnittstelle zwischen AmigaDos und der Diskette. Der Leser erfährt hier einiges über den Aufbau und interne Funktionen des AmigaDos. Die Autoren gehen auf die Organisation der Dateien auf der Diskette ein und besprechen den Aufbau der verschiedenen von AmigaDos verwendeten Blöcke auf der Diskette. Am Ende dieses Teils ist, wahrscheinlich noch in letzter Minute und der Aktualität wegen, ein kurzes Kapitel über Viren eingefügt worden, das jedoch vom Informationsgehalt her nicht dem restlichen Buch entspricht.

Der dritte und letzte Teil des Buches beschäftigt sich mit dem Kernstück der Diskettenprogrammierung - dem Trackdiskdevice. In diesen Kapiteln wird auf die Arbeitsweise und Fähigkeiten der im ROM enthaltenen Funktionen eingegangen. Der Leser erfährt hier über die Berechnung von Prüfsummen und den Ablauf der Device-Routinen viel Wissenwertes. Leider ist die Beschreibung der Disketten-Register (für direkten Hardware-Zugriff) nicht so gut gelungen wie die Beschreibung der Device-Routinen, so daß zum Verständnis der Funktionen dieser Register in vielen Fällen erst noch das Amiga Hardware Reference Manual herangezogen werden muß. Trotz allem bietet das Buch aber auch hier einen guten Einstieg in die Problematik.

Obwohl das Buch hier eigentlich schließt, verbergen sich in den Anhängen A bis C noch eine Menge Informationen, die gerade für die praktische Anwendung des besprochenen Stoffes nicht zu verachten sind. Es handelt sich hier um die Listings eines Diskettenmonitors, eines Floppybeschleunigers und eines Kopierprogramms. Leider ist die Dokumentation dieser Programme für die Zielgruppe des Buchs etwas zu schwach ausgefallen; auch wollte der Diskettenbeschleuniger erst mit der noch inoffiziellen Kickstart-Version 1.3 arbeiten.

Trotz einiger kleiner Haken und Ecken kann man dieses 59,- DM teure Werk weiterempfehlen. Das Buch bietet mit der die Listings enthaltenden Diskette eine Menge an Informationen und Anregungen, sich mit seinem Amiga und den Disketten auseinanderzusetzen. Merke: Es erforderte schon immer einiges, um die Probleme aus der Welt zu schaffen, die man vor der Anschaffung des Rechners nie auch nur hat erahnen können.

(GC)

ENDE



KICKS FÜR INSIDER

Wenn der GURU kommt, ist der Programmierer meist entsetzt, die Haare stehen ihm zu Berge, die Augen öffnen sich um einige Blenden und das Gebiß setzt an zum Angriff auf den meditierenden ´Freund´.
Wie man den GURU enttarnt und ihn verbannen kann, ist eines der Themen dieser KICKS.

INHALT

VON MARTIN WILHELM

GURU ADE, SCHEIDEN TUT (NICHT) WEH ...



Jeder, der auf dem Amiga selbst programmiert, kennt den berüchtigten "Software Failure"-Requester. Immer dann, wenn man durch einen Programmierfehler einen Prozessor-Trap erzeugt hat, kündigt dieser Requester eine "Guru-Meditation" an, was dann früher oder später unweigerlich mit einem Reset des Amiga verbunden ist.´

Diejenigen, die nicht glückliche Besitzer einer Harddisk sind, legen dann erst mal eine längere Zwangspause ein. Meistens hat man auch noch zu allem Überfluß umfangreiche Änderungen an einer Datei vorgenommen, die natürlich auf der RAM-Disk liegt (bzw. lag). Das hier vorgestellte Programm "GuruEx" schafft Abhilfe. Nach einem Trap geht es ganz normal weiter, und der Guru hat das Nachsehen. Um zu verstehen, wie man dem Guru ein Schnippchen schlagen kann, wird zu-

nächst erläutert, was beim Auftreten eines solchen Fehlers passiert.
Der Prozessorchip, der im Amiga Verwendung findet, bietet dem Programmierer die Möglichkeit, auf Fehlersituationen geordnet zu reagieren. In Fehlersituationen, die eine besondere Behandlung erfordern, wird das laufende Programm unterbrochen und eine Ausnahmebehandlung eingeleitet. Man bezeichnet diese Unterbrechung des laufenden Programms als Trap (engl.: Falle). Es können folgende "Fallen" auftreten:

- Bus Error.

Dieser Fehler kann im Prinzip nur durch externe Hardware-Erweiterungen verursacht werden. Da in diesem Fall meistens Hopfen und Malz verloren ist, wollen wir uns in diesem Zusammenhang nicht weiter darum kümmern. Häufiger ist schon ein

- Address Error.

Der tritt immer dann auf, wenn auf ein Wort oder Doppelwort mit ungerader Adresse zugegriffen wird. Dies kann sowohl den Zugriff auf einen Maschinenbefehl als auch auf einen Operanden bedeuten. Mit anderen Worten: Programmierfehler!

- Illegal Instruction.

Infolge eines Fehlers läuft das Programm Amok (z. B. Sprung in einen Datenbereich). Es soll ein Befehl ausgeführt werden, den der Prozessor nicht kennt. Was tun? Eine Sonderbehandlung ist fällig.

- Zero Divide.

Bekanntlich ist die Division durch Null nicht zulässig. Durch nachlässige Programmierung kann es vorkommen, daß trotzdem durch Null dividiert wird. Das Ergebnis ist ein Prozessor-Trap.

- CHK Instruction.

Der Befehlssatz des 68000-Prozessors ist insbesondere auch für den Einsatz von höheren Programmiersprachen konzipiert worden. Zur Prüfung, ob Anfangs- und Endwert von Feldern eingehalten werden, wird der Befehl "CHK" zur Verfügung gestellt. Werden die vorgegebenen Feldgrenzen überschritten, ist ein Trap fällig.

- TRAPV Instruction.

Ein unkontrollierter Überlauf bei arithmetischen Operationen ist selten erwünscht. Der 68000er stellt mit dem TRAPV-Befehl die Möglichkeit zur Verfügung, bei Auftreten eines Überlaufs eine Sonderbehandlung einzuleiten.

- Privilege Violation.

Der 68000-Prozessor kennt zwei Betriebsarten: den User Mode und den Supervisor Mode. Im User Mode sind einige Befehle nicht erlaubt. Dadurch wird erreicht, daß Befehle, die übergreifende Funktion haben (z. B. STOP [Anhalten des Prozessors]), nicht von jedem x-beliebigen Programm verwendet werden können. Benutzt ein Programm, das im User Mode läuft, einen privilegierten Befehl, resultiert daraus dieser Trap.

- Trace.

Ein Programm im Supervisormodus ist in der Lage, das Tracebit im Statusregister zu setzen. Ist das Tracebit gesetzt, so wird nach der Abarbeitung eines jeden Befehls ein Trap erzeugt. Dadurch wird die Überwachung (Trace) von Programmen unterstützt.

- Line 1010 Emulator.
- Line 1111 Emulator.

Dies sind im Prinzip ebenfalls nicht implementierte Befehle. Diese Befehlscodes werden jedoch mitunter verwendet, um z. B. Gleitkomma-Operationen auszuführen. Setzt man später einen Arithmetik-Prozessor in das System ein, übernimmt dieser die Ausführung der Gleitkommaberechnungen. Beide Traps liefern also ein Hilfsmittel zur Emulation (Nachbildung per Software) von Hardwarezusätzen. Für den normalen Amiga-Anwender bedeutet das Auftreten dieses Traps meistens, daß ein Programm in einen Datenbereich gesprungen und zufällig auf ein Datenwort getroffen ist, das mit $F0 oder $A0 beginnt.
Traps mit höheren Nummern sind entweder reserviert für spätere Erweiterungen oder gehören zu Interrupts. Diese sollen hier nicht weiter betrachtet werden.

Was passiert nun, wenn ein solcher Prozessor-Trap ausgelöst wird? Der Prozessor schaltet beim Erkennen eines Traps in den Supervisormodus um. Der Supervisormodus unterscheidet sich vom Usermodus dadurch, daß einige privilegierte Befehle (z. B. der

```
Guru Ex Dump
Trap Nr. 5: Division durch Null

Prozessor Status: PC=0008bc40 (00000010), SP=000a0bf0, CCR=--Z--

Register Inhalt:
D0=0000000a, D1=00000000, D2=00000fa0, D3=00000fa8
D4=00000001, D5=0009c4e0, D6=00000000, D7=0009be44
A0=00092ee6, A1=0009c49c, A2=00093194, A3=0008bc2c
A4=000a0bf8, A5=00ff44b4, A6=00ff44a8

Stackinhalt nach Trap:
5555  7777  00ff  4cae  0000  0fa0  0009  f086
000a  4c50  0000  0a20  0d01  000a  1612  0006
ffff  0000  ffff  0000  0100  0000  0000  0000
0000  8000  0000  0000  0000  00fc  2f98  0000
```

Die Dump-Ausgabe nach einem Guru

STOP-Befehl, der den Prozessor anhält) benutzt werden können. Außerdem wird ein anderer Stack benutzt. Nach dem Auftreten eines Traps sind des weiteren alle Interrupts gesperrt. Der Inhalt des Befehlszählers (PC) und das Statusregister werden auf den Stack gerettet. Anschließend wird in Abhängigkeit von der Ursache der Unterbrechung die Adresse des sogenannten Exception Vectors vom Prozessor ermittelt. Für jeden Trap ist ein solcher Vektor vorgesehen. Diese Vektoren sind nichts weiter als feste Plätze im Arbeitsspeicher des Rechners. Sie befinden sich in den ersten 1024 Byte des Arbeitsspeichers, also auf den Adressen 0 bis $3FF. Der Prozessor lädt den Befehlszähler mit dem Inhalt des für den Trap "zuständigen" Vektors, was einem Sprung auf diese Adresse gleichkommt. Hier nun kommt das Betriebssystem (Exec) des Amiga zum Zuge. Die Nummer des Trap wird ermittelt und als Langwort ebenfalls auf dem Stack abgelegt. Aus der Kontrollstruktur des Tasks, der diesen Trap verursacht hat, entnimmt Exec nun die Adresse des für die Behandlung von Traps zuständigen Programmteils und verzweigt dort hin. Normalerweise hat dies dann den berühmten Requester "Software Failure. Task held..." zur Folge, der meist nur Vorbote eines Guru ist. Legt man nun aber in der Kontrollstruktur des Task die Adresse einer eigenen Trapbehandlung ab, so kann man nicht nur das fehlerhafte Programm geordnet abbrechen (ohne eine Guru-Meditation!), sondern darüber hinaus noch zusätzliche Informationen bereitstellen, die die spätere Fehlersuche unterstützen.

Dieser Weg wurde mit dem Programm GuruEx beschritten. GuruEx installiert ein kleines Programm im Speicher des Amiga, das im Falle einer Trapbehandlung aufgerufen wird. Im ersten Teil des Programms wird der augenblickliche Zustand des Prozessors festgehalten. Der Inhalt aller Register wird zunächst im Arbeitsspeicher abgelegt. Bei diesem Programmteil muß insbesondere beachtet werden, daß es im Supervisormodus abläuft und alle Interrupts gesperrt sind. Es sollte daher möglichst kurz gehalten werden. Die Bereitstellung weiterer Informationen wird dementsprechend nach der Rückkehr in den User Mode durchgeführt.

```
            TTL       Guru Ex

*           Guru Ex trap handler   V 1.2
*           (nie wieder Gurus !)
*           M. Wilhelm
*           10-Apr-88      Version 1.2
*           (c) KICKSTART 1988

            INCDIR    "df1:include/"                  für DevPac Assembler

*           Include Dateien

            INCLUDE "exec/types.i"
            INCLUDE "exec/execbase.i"
            INCLUDE "exec/tasks.i"
            INCLUDE "exec/memory.i"
            INCLUDE "libraries/dos.i"
            INCLUDE "libraries/dosextens.i"
            INCLUDE "intuition/intuition.i"

*           Macro-Deklarationen

Call        MACRO
            jsr       _LVO\1(a6)
            ENDM

*           Exec und DOS - Routinen bekannt machen

      INCLUDE     "exec/exec_lib.i"           mit DevPac so schneller
      INCLUDE     "libraries/dos_lib.i"
      INCLUDE     "intuition/intuition_lib.i"

AbsExecBase         EQU      4                        Adresse von Execbase
LF                  EQU      10                       LineFeed

*           Teil 1 des Programms: Installation bzw. Entfernen von Guru Ex

_main       ;Start des Programms
            move.l    AbsExecBase,a6                  Basisadresse von Exec
            lea       DOSName(pc),a1                  LibName der DOS-Library
            moveq     #0,d0                           keine bestimmte Version
            Call      OpenLibrary                     DOS-Lib eröffnen
            move.l    d0,DOSBase                      DOS ist immer da, Fehler sind
                                                      ;einfach nicht erlaubt

            move.l    ThisTask(a6),a4                 Basisadresse der Task-Struktur
            move.l    TC_TRAPCODE(a4),a3              Adresse der Trap-Routine holen
            cmp.l     2(a3),a3                        ist es die eigene Traproutine ?
            beq.s     Remove                          falls ja, Trap entfernen

*           Speicher für den residenten Teil von Guru Ex besorgen

            move.l    #MEMF_PUBLIC,d1                 Exec soll Speicher 'festnageln'
            move.l    #TC_size,d0                     Größe des Traphandlers in Byte
            Call      AllocMem                        Speicher besorgen
            tst.l     d0                              gab es Speicher ?
            beq.s     NoMemory                   kein Speicher da (trübe Aussichten)
            move.l    d0,OwnAddr                 Kennung, daß GuruEx installiert ist
            move.l    a3,OldTrap                 Adresse der alten Traproutine merken
            move.w    #TC_size/2-1,d1                 Zähler für dbra
            move.l    d0,a1                           Zieladresse für Speichermove
            lea       TC_start(pc),a0                 Beginn des Codeteils
cploop      move.w    (a0)+,(a1)+                     wortweise kopieren
            dbra      d1,cploop                       Schleifenzähler
            move.l    d0,TC_TRAPCODE(a4)              neue Traproutine einklinken

*           Die Arbeit ist getan. Den Anwender informieren.

            lea       InstMsg(pc),a3                  Adresse der Message
            move.l    #IMsgLng,d3                     Länge der Nachricht
quit1       moveq     #0,d4                           Errorcode setzen (kein Fehler)
quit2       move.l    DOSBase,a6                      für die Benutzung von AmigaDOS
            Call      Output                          FileHandle für Screen Output
            move.l    d0,d1                           nach d1 für Write
            move.l    a3,d2                           Adresse des Textes nach d2
            Call      Write
            move.l    d4,d1                           Returncode holen
            Call      Exit                            das war's wohl

*     Guru Ex soll entfernt werden (eine verhängnisvolle Entscheidung !)

Remove      move.l    a3,a1                           Vorbereitung für FreeMem
```

Da die auf dem Stack abgelegte Rücksprungadresse zu dem den Trap erzeugenden Programm gehört, wird diese gegen die Adresse des zweiten Teils von GuruEx ausgetauscht. Dabei muß beachtet werden, daß der Prozessor bei einem Adress-Fehler zusätzlich zu Befehlszähler und Statuswort noch vier weitere Worte auf dem Stack ablegt. Der Sprung in den zweiten Teil von Guru Ex muß mit dem Befehl "RTE" erfolgen, nachdem der Stackpointer entsprechend korrigiert worden ist. In diesem zweiten Programmteil werden alle Register sowie die letzten 32 Worte des Stacks in einer Datei auf der RAM-Disk abgelegt und dann ein Requester auf dem Bildschirm ausgegeben. Aus der Fehlermeldung im Requester ist zu entnehmen, welcher Trap den Abbruch des Programms verursacht hat. Nach dem Anklicken erfolgt die Übergabe der Kontrolle an das CLI.

Vorsicht!

Bei aller Freude über die Aussicht auf ein "gurufreies" Programmiererleben sind jedoch einige Warnungen bzw. Hinweise anzubringen. GuruEx beseitigt nur Guru-Meditationen, die Folge von Prozessor-Traps sind. Abstürze des Betriebssystems, die z.B. durch Speichermangel oder fehlerhafte Aufrufe von Systemroutinen verursacht worden sind, bleiben von GuruEx (leider) unberührt. Zum Zweiten ist zu beachten, daß das Programm in der hier vorliegenden Form nur auf Amigas mit einem 68000er Prozessor arbeiten kann. Wurde der Amiga mit einem 68010- oder einem 68020-Prozessor ausgestattet, so muß bei der Behandlung von Adressfehlern das andere Datenformat des Stackframe (Zusatzinformationen auf dem Stack) dieser Prozessortypen berücksichtigt werden. Drittens werden geöffnete Dateien wegen des ungeplanten Programmabbruchs nicht ordnungsgemäß geschlossen, und eventuell mit der Systemroutine AllocMem() angeforderte Speicherbereiche werden nicht wieder an das Betriebssystem zurückgegeben. Schließlich muß man bedenken, daß GuruEx nur in dem CLI-Task wirksam ist, in dem es installiert worden ist. Verwendet man eine Erweiterung, die als neuer Task gestartet wird (wie z. B.

```
            move.l  #TC_size,d0             Länge des Speichers
            Call    FreeMem                 Speicher zurückgeben
            move.l  #RMsgLng,d3             Länge der Nachricht
            lea     RemMsg(pc),a3           Adresse der Message
            bra.s   quit1

*    Wenn es diese lächerlichen paar Bytes nicht mehr verfügbar sind,
*    dann gute Nacht. Sagen wir es trotzdem weiter ...

NoMemory            ;es gibt kein Bier auf Hawaii ... (Memory = Bier)
            move.l  #NMemLng,d3             Länge der Nachricht
            lea     NoMem(pc),a3            Anfangsadresse der Nachricht
            moveq   #20,d4                  dies ist ein Fehlercode
            bra.s   quit2                   zur Ausgabe

*           Datenbereich für den Installationsteil

DOSName dc.b 'dos.library',0

InstMsg dc.b  LF,LF,'***   Guru Ex V1.2 installiert    ***',LF,LF

RemMsg  dc.b  LF,LF,'***   Guru Ex entfernt    ***',LF,LF

NoMem   dc.b  LF,LF,'###   Kein Speicher frei für Guru Ex    ###',LF,LF

EndInst
            CNOP    0,4                     Den Rest auf Langwortgrenze
IMsgLng EQU         RemMsg-InstMsg          Länge der Insert-Nachricht
RMsgLng EQU         NoMem-RemMsg            Länge der Remove-Nachricht
NMemLng EQU         EndInst-NoMem           Länge der NoMemory-Nachricht


*   Hier beginnt der residente Teil der Trap-Routine.
*   Da dieser Teil im Supervisor Modus abläuft und sämtliche
*   Interrupts abgeklemmt sind, wird nur das allernötigste
*   getan und dann in den Usermodus zurückgekehrt.
*   Alle Register und die letzten 32 Worte auf dem Stack werden
*   in eine Datei geschrieben. Dieser Weg wurde gewählt, da
*   die Geschichte auch funktionieren soll, wenn der Trap nicht
*   durch eine CLI-Task generiert worden ist.
*   Man muß außerdem berücksichtigen, daß ausschließlich (!)
*   position-independent code verwendet werden muß, da kein hilfreicher
*   AmigaDOS-Lader das Geschäft einer Adresstranslation übernimmt.

TC_start            ;hier geht das nun aber los
            bra.s   TStart          Datenbereich überspringen
OwnAddr dc.l        0               eigene Startadresse als Kennung
OldTrap dc.l        0               originale Trapadresse
DOSBase dc.l        0               DOS-Base für späteren Gebrauch

*           Als Erstes stellen wir fest, was es für ein Trap war.
*    Die Traps Nummer 3 (Bus Fehler, ganz böse !), 9 (Trace, da
*    wird wohl ein Monitor im Spiel sein) sowie 12 bis ...
*    (Trap-Befehle und Interrupts) werden von GuruEx nicht behandelt,
*    da sie entweder fatal sind (Bus-Fehler) oder wahrscheinlich vom
*    Anwender so gewollt (Trace, Trap-Befehle).

TStart  cmp.l       #3,(sp)         war es etwa ein Bus Error ?
        bmi.s       ExecTrap        damit soll sich Exec ärgern
        beq.s       AddrErr         Adressfehler hat anderen Stackframe
        cmp.l       #12,(sp)        Test auf Interrupts und TRAP's
        bpl.s       ExecTrap        auch die sind ein Fall für Exec
        cmp.l       #9,(sp)         wenn es ein Trace-Trap war,
        beq.s       ExecTrap        muß sich eine Traceroutine kümmern

*           hier müssen wir nun ran
*            (der 68000 ist schon echt gut ! Alle diese Spielchen ohne ein
*             einziges Register verändern zu müssen !)

            lea     TrapCode(pc),a0         Adresse für Savebereich
            move.w  2(sp),(a0)+             die Nummer des Traps merken
saveall     move.w  4(sp),(a0)+             Statusregister merken
            move.l  6(sp),(a0)+             PC des Fehlers
            movem.l d0-d7/a0-a6,(a0)        alle Register merken
            lea     UserSt(pc),a0           Startadresse für den Userteil
            move.l  a0,6(sp)                auf den Stack schieben
            addq.l  #4,sp                   Stack bereinigen
            rte                             return from exception

*           Bei einem Address Error wird ein anderer Stackframe abgelegt.
*           Achtung!
*           Die Prozessoren 68010 und 68020 haben noch ein anderes Stack-
```

```
*       Frameformat. Dies muß exakt hier berücksichtigt werden!
*   Mangels Testmöglichkeit wird hier nur der "normale" Stackframe
*   für den 68000 behandelt.

AddrErr lea     TrapCode(pc),a0    Adresse für Savebereich
        move.w  2(sp),(a0)+        die Trapnummer merken
        addq.l  #8,sp             Stack nach Adress-Error korrigieren
        bra.s   saveall            alles andere retten

*   Hier landen alle die Traps, mit denen wir nichts anfangen können

ExecTrap        ;an die Original Exec Routine übergeben
        move.l  OldTrap(pc),-(sp)  die Adresse auf den Stack
        rts                        so tun, als wäre es eine Subroutine

*       Hier beginnt der Teil, der im User Mode abläuft

UserSt  lea     StackPtr(pc),a0          Adresse der Save-Area für SP
        move.l  sp,(a0)                  Stackpointer merken

*  Wenn es ein CLI-Prozess war (was es wohl sein sollte), können
*  wir zur Freude der Nachwelt noch die Anfangsadresse des geladenen
*  Code beisteuern. Daraus kann dann später (hoffentlich) die Adresse
*  des Befehls ermittelt werden, der den Trap verursacht hat.

        move.l  AbsExecBase,a6      Execbase besorgen
        move.l  ThisTask(a6),a0     Anfangsadresse des TCB
        move.l  pr_CLI(a0),a1       BPTR auf CLI holen
        beq.s   NoCLI              Es war keine CLI-Task !
        add.l   a1,a1              Dies alles nur, weil Commodore
        add.l   a1,a1               unbedingt BCPL benutzen musste!
        move.l  cli_Module(a1),a1   Das bringt uns einen BPTR (grr...)
        add.l   a1,a1               auf die Segmentliste des Programms
        add.l   a1,a1            a1 zeigt jetzt auf den nächsten BPTR
        addq.l  #4,a1                Hier ist die Startadresse
NoCLI   lea     FirstCode(pc),a0     Adresse besorgen
        move.l  a1,(a0)            und Anfang des Codesegments merken
                                         ;(=0 wenn keine CLI-Task)

*       Jetzt machen wir etwas leicht Anrüchiges:
*       Wir holen die DOS-Base nicht über ein OpenLibrary sondern
*       durch einen zuvor ermittelten Wert. Dies ist aber nicht so
*       abwegig, da DOS eine residente Library ist, die also nicht
*       zugeladen wird. Daher kann man mit einiger Sicherheit davon
*       ausgehen, daß DOS immer an der gleichen Adresse liegt.

        move.l  DOSBase(pc),a6      DOSBase nach A6
        lea     DumpFN(pc),a1       Name des Dumpfile
        move.l  a1,d1               für Open
        move.l  #MODE_NEWFILE,d2
        Call    Open                Dumpfile eröffnen
        tst.l   d0                  war es möglich ?
        bne.s   DumpIt              wenn nicht
        lea     NoDumpFile(pc),a1   Adresse des 'Ich bedaure'-Textes
        bra.s   StoreH

*   Jetzt wird erst einmal alles, was gerettet wurde, in einer
*   Datei (auf RAM:) gespeichert.

DumpIt  move.l  d0,d1                    FileHandle für Write
        lea     TrapCode(pc),a0
        move.l  a0,d2                    Pufferadresse
        move.l  #EndOfAll-TrapCode,d3    Länge nach d3
        move.l  d1,-(sp)                 FileHandle auf Stack
        Call    Write                    Register in File ablegen
        move.l  (sp),d1                  FileHandle vom Stack holen
        move.l  StackPtr(pc),d2  alten Stackpointer als Pufferadresse
        move.l  #64,d3          die letzten 32 Worte werden gerettet
        Call    Write                    auch in die Datei schreiben
        move.l  (sp)+,d1                 FileHandle vom Stack holen
        Call    Close                    Datei schliessen

*   Alles gerettet. Nun müssen einige Adressen in den
*   IntuiText-Strukturen für den Requester eingebaut werden.

        lea     HText(pc),a1             Adresse Help-Text
StoreH  lea     HelpMsg(pc),a4           Adresse in IntuiText
        move.l  a1,(a4)                  in IntuiText-Struktur ablegen

*   Intuition eröffnen

Request move.l  AbsExecBase,a6           ExecBase besorgen
```

die MCC-Shell), so muß GuruEx unter diesem Task installiert werden. Wenn man also GuruEx in die Startup-Sequence einbindet, muß man es an einer Stelle tun, an der der aktuelle CLI-Task bereits aktiv ist.

GuruEx vom CLI...

Guru Ex wird einfach dadurch aktiviert, daß man in dem CLI-Task, den man von Gurus befreien möchte, GuruEx (ohne Parameter) aufruft. Für alle diejenigen, die ohne den Guru nicht leben können, gibt es die Möglichkeit, GuruEx aus dem System zu entfernen. Ein zweiter Aufruf von GuruEx erledigt dies.

Allen leidgeprüften Programmierern, die ihren eigenen Fehlern auf die Spur kommen wollen, stellt GuruEx zusätzliche Informationen zur Verfügung, die bei der Fehlersuche sehr hilfreich sein können. Wie zuvor schon erwähnt wurde, wird der Zustand des Prozessors zum Zeitpunkt des Traps in einer Datei gesichert. Der Einfachheit halber wurde ein fest in GuruEx einprogrammierter Dateiname verwendet. Diese Informationen werden auf der RAM-Disk abgelegt. Die Datei heißt Guru.dmp, hat eine Länge von 140 Byte und ist folgendermaßen aufgebaut:

Wort Nr.0: Nummer des Trap

Wort Nr.1: Status des Prozessors (Condition Codes)

Wort Nr.2: Befehlszähler bei Trap. Der Befehlszähler zeigt auf den Befehl nach der Fehlerursache (außer bei einem Address Error)

Wort Nr. 3: Inhalt von D0

⋮

Wort Nr. 17: Inhalt von D7

Wort Nr. 19: Inhalt von A0

⋮

Wort Nr. 31: Inhalt von A6

Wort Nr. 33: Inhalt des Stackpointers

Wort Nr. 35: Anfangsadresse des Programmcodes

Wort Nr. 37: Inhalt des Stacks

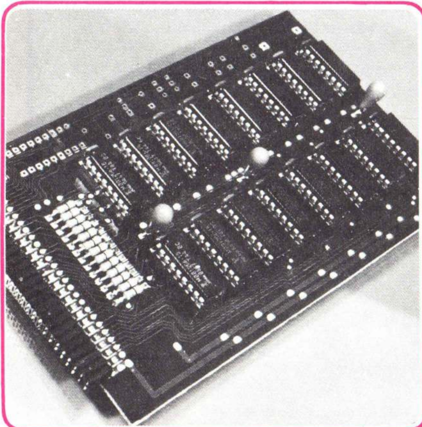
⋮

Wort Nr. 69: Inhalt des Stack

ausgereifte Ingenieurleistung ● 14 Tage Umtauschrecht ● 2 Jahre Garantie ● fast alle IC'S gesockelt ● nur professionelle Leiterplatten ● Bauteile namhafter Hersteller ● mit Bedienungsanleitung ● Blockschaltbild ● teilweise Schaltplan

Frischer Wind auf dem Hardwaresektor

ALCOMP
COMPUTERHARDWARE



500er Speichererweiterung

Für 512k zusätzliches RAM ● alle RAM's gesockelt ● selbstkonfigurierend ● abschaltbar ● ● Uhrenschaltung auf Platine mit Akku- bzw. Batteriepufferung nachrüstbar

Komplett mit 512k	Preis auf Anfrage
Superpreis mit Uhr	Preis auf Anfrage
Bauteilesatz für Uhr ohne Akku	24.-
Leerplatine mit Stecker	*39.-

*mit Schaltplan und Bestückunsliste



Profilaufwerk 3,5"

Metallgehäuse ● einstellbare Laufwerknummer mit Displayanzeige ● digitale Trackanzeige ● ● Write Protect am Laufwerk schaltbar ● abschaltbar ● durchgeschleifter Bus
1 Jahr Garantie

SuperALCOMPreis 349.-

Trackanzeige

Für DFO-DF3 einstellbar ● für alle Laufwerke (3,5"/5,25") ● Laufwerkbus durchgeschleift ● mit Gehäuse

SuperALCOMPreis 49.-

Wir suchen ständig Hardware-Entwicklungen. Wir garantieren gute Umsatzprovisionen und ehrliche Abrechnung

3,5" Laufwerk

Für alle Amiga's ● einstellbare Gerätenummer ● ● abschaltbar ● Metallgehäuse ● superflach ● 1 Zoll (2,54 cm) ● durchgeschleifter Bus ● TEAC Laufwerk
1 Jahr Garantie

komplett anschlußfertig 279.-

Laufwerk 5,25"

40/80 Track ● Laufwerksbus durchgeschleift ● ● abschaltbar ● einstellbare Adressen ● MS-DOS-kompatibel ● mit Diskchange
1 Jahr Garantie

SuperALCOMPreis	339.-
HD 1,6 MB (umschaltbar)	359.-

Gemischtes Doppel 3,5/5,25"

einzeln ein-/abschaltbar ● einstellbare Laufwerksnummern mit Anzeige ● durchgeschleifter Bus ● bei 5,25" 40/80 Tracks umschaltbar ● Metallgehäuse
1 Jahr Garantie

SuperALCOMPreis 598.-

Basislaufwerke

1 Jahr Garantie

TEAC FD 135 FN 3,5" 1MB superslimline	218.-
TEAC FD 55 FR 5,25" 40/80 Tracks 1MB	239.-
TEAC FD 55 FR 5,25" 40/80 Tracks 1,6MB Diskchange	259.-
3,5" Gehäuse	25.-
5,25" Gehäuse	25.-
Gehäuse für "Gemischtes Doppel"	45.-

Laufwerkanschlußkabel

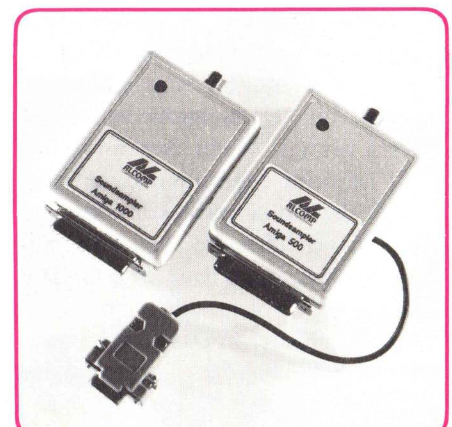
Zum Anschluß von Laufwerken an alle Amiga's ● ● mit Ansteuerelektronik

Für 3,5" Laufwerk	39.-
Für 5,25" Laufwerk	49.-

Steckplatzerweiterung 3-fach für Laufwerke

Jeder Steckplatz abschaltbar und einstellbare Laufwerksnummer ● Steckplatzerweiterung direkt am Amigagehäuse ● Dadurch keine Kabellängenprobleme

Anschlußfertig zum SuperALCOMPreis 49.-



Soundsampler

Für Amiga 1000 und 500 mit Software ● Type bei Bestellung bitte angeben ● 8-Bit Datenbreite ● Betrieb am Parallelport (Druckerport) ● Mit Vorverstärker für Micro-Anschluß (Cinch-Buchsen) ● Musik- und Sprachdigitalisierung möglich ● Arbeitet mit fast allen Digitizer-Programmen ● Formschönes Gehäuse

SuperALCOMPreis 79.-

MIDI - Interface

4 Kanäle einschließlich 1 Thru ● Optische Datenanzeige ● Formschönes Gehäuse

Wahnsinnspreis von nur 89.-

Bootselector 19.90

Kickstartumschaltung

Bauen Sie die anderen Kickstart-Versionen in Ihren Amiga 500 ● Einfacher Einbau ohne Löten ● für Original-Kickstart-ROM und 2 zusätzliche Versionen auf EPROM ● EPROM-Programmierservice auf Anfrage

SuperALCOMPreis 59.-

Public-Domain-Disketten

Große Auswahl ● schnelle Auslieferung

Einzelstück 6.-

Staffelpreise auf Anfrage

Bestellung und Versand

ALCOMP
A. Lanfermann
Lessing Str. 46
5012 Bedburg
Tel. 0 22 72/15 80

Nachnahmeversand NN-Spesen 7.50 DM b. Vorkasse 3.- DM. Auslandsbestellungen: Nachnahmeversand NN-Spesen 10.- DM b. Vorkasse 5.-DM. Wir liefern Ihnen auf Ihre Rechnung und Gefahr zu den Verkaufs- und Lieferbedingungen des Elektronikgewerbes. Postgiroamt Köln (BLZ 370 100 50) 275 54-509

PLUM Entwurf + Druck · 5162 Düren-Niederzier · Tel. 0 24 28/711

Analyse gefällig?

Zur bequemen Auswertung dient das Programm "GXDA" = GuruEx Dump Analyse (dump: engl. für Abfallhaufen). GXDA bringt den "Abfall", den das abgestürzte Programm in der Datei RAM:Guru.dmp hinterlassen hat, in Klarschrift auf den Bildschirm. Man ruft nach einem Fehler einfach das Programm GXDA auf. GXDA benötigt keine weiteren Angaben. Ist keine Datei Guru.dmp in der RAM-Disk abgelegt, gibt das Auswertungsprogramm eine entsprechende Fehlermeldung aus. Es wird außerdem geprüft, ob die Dateigröße exakt 140 Bytes beträgt. Wenn dies nicht der Fall ist, wird angenommen, daß der Inhalt der Datei keine korrekten Informationen über die Registerstände usw. enthält. Auch dann wird GXDA mit einer Fehlermeldung beendet. War jedoch alles in Ordnung, so wird die Ursache für den Trap wird im Klartext angezeigt. Die außerdem angezeigten Registerinhalte bedürfen wohl keiner weiteren Erläuterung. Zu einigen Anzeigen ist jedoch eine zusätzliche Erklärung vonnöten. Da ist zunächst das Condition Code Register (CCR). In diesem Register wird das Ergebnis des zuletzt ausgeführten Befehls festgehalten. Es gibt fünf dieser "Bedingungen":

X = extend. Bei Rechenoperationen, die mit sehr großen Zahlen arbeiten, zeigt dieses Bit einen Übertrag an, der bei Rechenoperationen mit höherer Genauigkeit benutzt wird.

N = negative. Wenn das Ergebnis einer Rechenoperation negativ war, ist dieses Bit gesetzt.

Z = zero. Gesetzt, wenn das Ergebnis gleich Null ist.

V = overflow. Das Ergebnis einer Rechenoperation hatte einen Überlauf.

C = Übertrag.

Ein Minuszeichen zeigt an, daß das entsprechende Bit im CCR nicht gesetzt ist.

Der Befehlszähler (PC = Program

```
             moveq    #0,d0                     Version egal
             lea      IntuiName(pc),a1          Name der Intuition-Lib
             Call     OpenLibrary               LibraryBase holen
             tst.l    d0                        konnten wir Intuition eröffnen ?
             beq.s    Fatal                     wenn nicht: Au weia !
             move.l   d0,a6                     für späteren Aufruf

*    Vorbereitung für Requester
*    Für den aufgetretenen Fehler den zutreffenden Text ermitteln

             moveq    #0,d0                  d0 vorlöschen
             move.w   TrapCode(pc),d0        Nummer des Traps
             subq.w   #3,d0                  für Offset-Berechnung
             lea      TOffset(pc),a0         Adress-Offset für Zugriff auf Text
             add.l    d0,a0                  für den richtigen Trap-Text
             move.b   (a0),d0                Offset nach d0
             lea      TrapMsg(pc),a0         Adresse der Texttabelle
             add.l    d0,a0                  jetzt haben wir die Adresse in a0
             lea      HelpText(pc),a4           Adresse in IntuiText
             move.l   a4,a3                     merken
             move.l   a3,-(a4)                  Verzeigerung aufbauen
             move.l   a0,-(a4)                  Adresse des Fehlertextes
             lea      ErrText(pc),a4            für weitere Verzeigerung
             move.l   a4,a3
             move.l   a3,-(a4)                  Zeiger einbauen
             lea      BText(pc),a3              Pointer auf BodyText
             move.l   a3,-(a4)                  einsetzen
             lea      CanText(pc),a4         da wir nicht wissen, wo wir sind
             lea      CText(pc),a0              Adresse des Textes
             move.l   a0,(a4)                   in IntuiText-Struktur

*    Nun können wir endlich den Requester steigen lassen

             moveq    #0,d0                     keine zusätzl. Flags
             move.l   d0,d1
             move.l   #240,d2                Breite des Requesters (240 Pixel)
             moveq    #70,d3                    und die Höhe (70 Pixel)
             lea      BodyText(pc),a1           Haupttext des Requesters
             lea      CancelText(pc),a3         Text um zu quittieren
             move.l   d0,a2                     kein positiver Text
             move.l   d0,a0                     eigenes Window
             Call     AutoRequest               Requester zeigen

*    Um das Ergebnis brauchen wir uns nicht zu kümmern

Fatal        moveq    #20,d1                   es war in jedem Fall fatal
             move.l   DOSBase(pc),a6           nun ist wieder Exec am Drücker
             Call     Exit                     geschafft. Guru ausgetrickst.

*    IntuiText - Strukturen für den Requester
*    Die Strukturen werden dynamisch mit den Adress-Zeigern versehen

BodyText              dc.b     0,1,RP_JAM2
                      dc.w     30,4
                      dc.l     0,0,0

ErrText               dc.b     0,1,RP_JAM2
                      dc.w     30,14
                      dc.l     0
                      dc.l     0,0

HelpText              dc.b     0,1,RP_JAM2
                      dc.w     30,24
                      dc.l     0
HelpMsg               dc.l     0,0

CancelText            dc.b     0,1,RP_JAM2
                      dc.w     6,3
                      dc.l     0
CanText               dc.l     0,0

BText                 dc.b     'Software Failure:',0
                      CNOP     0,2

HText                 dc.b     'Run GXDA to debug.',0
                      CNOP     0,2

NoDumpFile            dc.b     'Sorry, no dumpfile!',0
                      CNOP     0,2

CText                 dc.b     'Cancel',0
```

```
*   Texte für die diversen Traps

                   CNOP   0,2
TrapMsg            dc.b   'Address Error',0
                   CNOP   0,2
TM4                dc.b   'Illegal Instruction',0
                   CNOP   0,2
TM5                dc.b   'Zero Divide',0
                   CNOP   0,2
TM6                dc.b   'CHK Instruction',0
                   CNOP   0,2
TM7                dc.b   'TRAPV Instruction',0
                   CNOP   0,2
TM8                dc.b   'Privilege Violation',0
                   CNOP   0,2
TM9                dc.b   'Trace',0          kommt nicht vor
                   CNOP   0,2
TM10               dc.b   'Line 1010 Emulator',0
                   CNOP   0,2
TM11               dc.b   'Line 1111 Emulator',0

*   Tabelle mit Text-Offsets

TOffset            dc.b   0
                   dc.b   TM4-TrapMsg
                   dc.b   TM5-TrapMsg
                   dc.b   TM6-TrapMsg
                   dc.b   TM7-TrapMsg
                   dc.b   TM8-TrapMsg
                   dc.b   TM9-TrapMsg
                   dc.b   TM10-TrapMsg
                   dc.b   TM11-TrapMsg

*       Name der Datei, in der die Dumpinformation abgelegt wird

                   CNOP    0,2
DumpFN             dc.b    'ram:Guru.dmp',0

                   CNOP    0,2
IntuiName          dc.b   'intuition.library',0

*       Save Area
                   CNOP    0,2
TrapCode           ds.w    1                  hier wird der Trapcode abgelegt
Status             ds.w    1                  hier ist das Statusregister
TaskPC             ds.l    1                  PC nach Trap
Regs               ds.l    15                 Register D0-D7, A0-A6
StackPtr           ds.l    1                  Stackpointer
FirstCode          ds.l    1                  Anfangsadresse des Programms
EndOfAll           ;Ende von allem
TC_size EQU        EndOfAll-TC_start          Länge des residenten Teils
                   END
```

Listing 1: Der GURU EX Traphandler. Sichert vor einem Trap-Aufruf wichtige Register und den Stack. Der Guru ist besiegt.

```
/*    Dumpanalyse für Guru Ex.

      M. Wilhelm 1988
         V1.0
         Apr. 1988
      (C)KICKSTART 1988          */

#include "exec/types.h"
#include "libraries/dos.h"
#include "functions.h"

#define  BufLng    140
#define  ErrCode   20L

main()              /* Aufruf: gxda  */

{
   long  Buff[BufLng/4];
   long  error = 0L;
   static int   stcode [] = { 16, 8, 4, 2, 1 };
   static char  Flags[] = {88,78,90,86,67};  /* die Prozessor Flags */
   /* Die Prozessor Flags    X  N  Z  V  C  */

```

Counter) gibt die Adresse des Befehls an, der unmittelbar auf den den Trap verursachenden Befehl folgt. Da jedoch nicht von vorneherein feststeht, an welche Adresse ein Programm geladen wird, ist der Stand des Befehlszählers nur von eingeschränktem Wert. Kennt man jedoch die Anfangsadresse des Codeteils des betroffenen Programms, so ist daraus die Adresse des fehlerhaften Befehls in den meisten Fällen zu ermitteln. Das Analyseprogramm GXDA zeigt deshalb nicht nur den Befehlszählerstand an, sondern gibt auch den Inhalt des PC relativ zur Ladeadresse aus. Dieser Wert steht (in Klammern) hinter dem absoluten Wert des Befehlszählers. Zusammen mit den Inhalten der Register kann die Auswertung des "Abfalls" eine gute zusätzliche Hilfestellung beim Entwanzen (engl. bug : Wanze, to debug: entwanzen) darstellen. Auch der Inhalt des Stack kann wertvolle Hinweise zur Fehlersuche liefern. Da bei Unterprogrammsprüngen die Rücksprungadressen auf dem Stack abgelegt werden, läßt sich mit einigem Spürsinn nachvollziehen, welche Unterprogramme angesprungen wurden und mit welchen Parametern sie versorgt worden sind - falls es sich um ein C-Programm gehandelt hat. Eine solche Analyse ist natürlich nur sinnvoll, wenn man auch die Möglichkeit zur Korrektur hat, d. h., die Quelle des fraglichen Programms muß vorliegen. Alle die, die nicht selbst programmieren, benötigen das Auswertungsprogramm nicht.

GuruEx wurde mit dem Devpac-Assembler übersetzt. Bei einer entsprechenden Anpassung sollte auch die Verwendung anderer Assembler (z. B. Seka) ohne Probleme möglich sein.

Das Auswertungsprogramm GXDA wurde in C geschrieben. Zur Verwendung kam Aztec C-Compiler, wobei alle Versionen hierzu herangezogen werden könen (V 3.40, V 3.60).

Die Übersetzung mit dem Lattice-C-Compiler dürfte keine Schwierigkeiten bereiten, wenn man die Zeile

```
#include "functions.h"
```

löscht bzw. nicht erst mit abtippt. Mit Hilfe dieser Include-Datei werden lediglich die Aufrufe von Betriebssystemroutinen des Amiga als LONG de-

klariert, da Aztec standardmäßig wortweise arbeitet. Für die Übersetzung mit Aztec C sind die folgenden Aufrufe erforderlich:

```
cc gxda.c-s-i include -o ram:gxda.o
ln ram:gxda.o lib/c32.lib -o GXDA
delete ram:gxda.o
```

Es empfiehlt sich, GuruEx mit in die Startup-Sequence einzubinden, um von vornherein dem Guru das Leben etwas schwerer zu machen. Um auszuprobieren, ob und wie GuruEx funktioniert, soll zum Schluß noch ein ganz kurzes Assemblerprogramm gezeigt werden, das einen Prozessortrap erzeugt. Dies ist ganz einfach durch eine Division durch Null zu bewerkstelligen. Hier das Programm:

```
moveq   #10,d0
   den Wert 10 ins Register d0
moveq   #0,d1
   die Null nach Register d1
divu    d1,d0
   d0 = d0/d1
rts
   hier kommen wir nie hin !
END
```

Wird nun das Programm zero aufgerufen, ohne daß zuvor GuruEx installiert war, erscheint der anfangs erwähnte Requester. Wenn man nach dem Reset GuruEx aktiviert und zero erneut aufruft, passiert außer einer Fehlermeldung nichts.
Der Guru ist besiegt!

ENDE

```
     static char  *TrapMsg[] = {      /* die Fehlercodes für die Traps */
        "Bus Fehler",                           /* Bus Error */
        "Adress Fehler",                        /* Address Error */
       "Ungültige Instruktion",                 /* Illegal Instruction */
        "Division durch Null",                  /* Zero Divide */
        "CHK Befehl erkannt",                   /* CHK Instruction */
        "TRAPV Befehl ausgeführt",              /* TRAPV Instruction */
        "Privilegierter Befehl im User Modus",  /* Privilege Violation */
           "Trace",                             /* Tritt nicht auf */
           "Befehlscode 0xA.. erkannt",         /* Line 1010 Emulator */
           "Befehlscode 0xF.. erkannt" };       /* Line 1111 Emulator */

     int   i,tc,status;
     struct FileHandle *fh;          /* FileHandle für Dumpfile */

     /* Dumpfile eröffnen */
     if ((fh = Open("ram:Guru.dmp",MODE_OLDFILE)) == NULL)
     { error = IoErr();              /* kein Guru Dumpfile gefunden */
      printf("\n\n***Fehler bei Open Guru.dmp.Fehlernr. %ld\n\n",error);
       return(ErrCode);
     }

     if ((Read(fh,Buff,BufLng)) != BufLng)
     {  printf("\n\n*** Dumpfile hat nicht die richtige Länge.\n");
        printf("    Analyse des Dumps nicht möglich.\n\n");
        Close(fh);
        return(ErrCode);
     }

     Close(fh);  /* Jetzt haben wir alle Informationen */

     printf("\n\tGuru Ex Dump\n");
     tc=Buff[0] >> 16; status=Buff[0] & 0xffffL;
     printf("\tTrap Nr. %d: %s\n\n",tc,TrapMsg[tc-2]);
     printf("\tProzessor Status: ");

     /* Hier werden PC, SP und CCR ausgegeben */

     printf("PC=%08lx (%08lx), SP=%08lx,",Buff[1],Buff[1]-
                                      Buff[18],Buff[17]);
     printf(" CCR=");
     for (i=0; i < 5; i++)
     {
        if ((status & stcode[i]) == 0)
           putchar(45);  /* Minuszeichen, wenn Flag nicht gesetzt */
        else putchar(Flags[i]);  /* gesetzte Flags anzeigen */
     }

     /* Erst die Inhalte der Datenregister ausgeben */

     printf("\n\n\tRegister Inhalt:\n\t");

     for (i=2; i < 10; i++)
     {
        printf("D%d=%08lx",i-2,Buff[i]);
        if ((i==5) | (i==9)) printf("\n\t"); else printf(", ");
     }

     /* Jetzt sind die Adressregister dran */

     for (i=10; i < 17; i++)
     {
        printf("A%d=%08lx",i-10,Buff[i]);
        if ((i==13)|(i==16)) printf("\n\t"); else printf(", ");
     }

     /* Nun noch die letzten 32 Worten auf dem Stack */

     printf("\n\tStackinhalt nach Trap:\n\t");

     for (i=19; i < 35; i++)
     {
        printf("%04x  %04x ",(short)(Buff[i]>>16),
                   (short)(Buff[i]&0xffff));
        if (((i-19) & 3) == 3) printf("\n\t");
     }
     printf("\n");
     return(0L);
  }
```

Listing 2: GURUDUMP zeigt die geretteten Daten, die vor Auftreten des Fehlers gesichert wurden.

VON KNUT BUSSIAN

GROSSE AUSWAHL

Batchdateien per Menü



Während nun fast zweijähriger Benutzung des Amiga mußte ich immer wieder die Startup-Sequence umschreiben oder den Boot-Vorgang mit CTRL-D unterbrechen, falls ich an der Systemkonfiguration etwas ändern wollte. Allerdings ging mir dieser Zeitverlust recht schnell auf die Nerven. Aus diesem Grund ist "Witch-Batch" entstanden.

Das Programm sucht in einem vorgegebenen Verzeichnis nach Batch-Dateien, auch als Stapel- oder Befehls-Dateien bekannt. Damit es die Batch-Dateien auch als solche erkennt, muß man sie so umbenennen, daß sie auf ".bat" enden (z.B. "startup-sequence.bat"). Im Gegensatz zu manch anderem System erkennt der Amiga solche Dateien nicht. Der ursprüngliche Suchpfad ist "S:". Dieser Pfad wird ausgewählt, sobald "WitchBatch" ohne weitere

Parameter aufgerufen wird. Falls keine Batch-Dateien in einem vorgegebenen Verzeichnis gefunden wurden, meldet sich das Programm mit "Keine Batch-Dateien vorhanden" ab. Falls aber doch Batch-Dateien gefunden wurden, bekommt man eine tabellarische Übersicht mit einer Zahl und dem dazugehörigen Batch. Nun kann man entweder mit "0" das Programm beenden oder mit der Zahl die zugehörige Batch-Datei zur Ausführung bringen.

"WitchBatch" kann bis zu 20 Batch-Dateien in einem Verzeichnis verarbeiten. Wird eine größere Liste benötigt, kann man einfach MAXBATCH im Quellcode verändern. Dort kann man natürlich auch den voreingestellten Suchpfad ändern. Ich rufe "WitchBatch" als erstes Kommando in der Startup-Sequence auf. Dies ist wohl das einfachste und effizienteste Anwendungsbeispiel.

Sicherlich kann man das Programm noch verbessern. Ich wäre dankbar für jeden Tip, wie man z.B. "WitchBatch" durch sich selbst aufrufen kann, ohne daß man dabei in einer Endlosschleife stecken bleibt.

Mit diesem Programm können Sie sich, wie erwähnt, bis zu 20 verschiedene Batch-Dateien erstellen, die natürlich die unterschiedlichsten Aufgaben durchführen könen. Beispielsweise kann die eine Datei alle CLI-Befehle ins Ram kopieren, eine andere nur einzelne, während eine dritte gleich das Amiga-BASIC oder den PC-Treiber lädt.

```
/*
 * WitchBatch -> Programm zum selektiven Aufrufen
 * einer Batch-Datei
 * Autor: Knut Bussian
 * (c) KICKSTART 1988
 * Für Aztec C und den Lattice C
 * Programm findet sich, wie alle anderen auch,
 * in compilierter Form und als Source-Code
 * auf der Monatsdiskette
 */

#include <libraries/dos.h>
#include <exec/memory.h>

#define NOR 0
#define UNL 1
#define INV 2
#define CLS 3
#define MAXBATCH 20
#define MAX_PATH 50

char *scr[] =
{"[0m", "[4m", "[7m", "c"};

char batchlist[MAXBATCH][MAX_PATH]; /* Liste der
                                   gefundenen Batch Dateien */
char pfad[MAX_PATH] = "s:";       /* Default fuer Pfadname */

extern struct FileInfoBlock *AllocMem();
extern struct FileLock *Lock();

int searchbatch(pfad)
char pfad[MAX_PATH];
{
int i, k, anz;
struct FileInfoBlock *finfo;
struct FileLock *mylock;
char temp[MAX_PATH];
anz = 0;

finfo = AllocMem((long)sizeof(struct FileInfoBlock),
                         MEMF_CHIP | MEMF_CLEAR);
if(finfo == NULL)
        {
        printf("Kein Chip-Mem fuer FileInfoBlock!\n");
        exit();
        }
      else
        {
        mylock = Lock(pfad, ACCESS_READ);
        if(mylock == NULL)
                {
                printf("%s nicht gefunden!\n", pfad);
                FreeMem(finfo, (long)sizeof(*finfo));
                exit();
                }
        if(Examine(mylock, finfo) == NULL)
                {
                printf("DOS-Fehler #%d\n", IoErr());
                FreeMem(finfo, (long)sizeof(*finfo));
                exit();
                }
        while(ExNext(mylock, finfo) && anz < MAXBATCH)
           {
            if(finfo->fib_DirEntryType<0)
              {
               for(i=0;i<strlen(finfo->fib_FileName);i++)
                 {
                  if(finfo->fib_FileName[i] == '.')
                  if(finfo->fib_FileName[i+1] == 'b')
                  if(finfo->fib_FileName[i+2] == 'a')
                  if(finfo->fib_FileName[i+3] == 't')
                   {
                    strcpy(batchlist[anz],
                           finfo->fib_FileName);
                    anz++;
```

```
                        }
                    }
                }
            }
        FreeMem(finfo, (long)sizeof(*finfo));
        UnLock(mylock);
        }
/* Sortieren */

for(i=0;i<anz-1;i++)
    {
      for(k=i+1;k<anz;k++)
        {
        if(batchlist[i] > batchlist[k])
          {
           strcpy(temp, batchlist[k]);
           strcpy(batchlist[k], batchlist[i]);
           strcpy(batchlist[i], temp);
          }
        }
    }
return(anz);
}

void main(argc, argv)
int argc;
char *argv[];
{
int anz, i, which;
char command[MAX_PATH];
if(argc == 2)
  {
   if(argv[1][0] == '?')
     {
      printf("%sWitchBatch%s von Knut Bussian (1988)\n",
              scr[UNL], scr[NOR]);
      printf("Voreingestellter Suchpfad:  %s %s %s\n\n",
              scr[INV], pfad, scr[NOR]);
      exit();
     }
  else
  strcpy(pfad, argv[1]);
  }
anz = searchbatch(pfad);

wahl:
if(anz == 0)
        {
        printf("Keine Batch Dateien vorhanden!\n");
        exit();
        }
printf("%s",scr[CLS]);
printf("\n\n\t%sVorhandene Batch Dateien:%s\n\n",
        scr[UNL], scr[NOR]);
for(i=0;i<anz;i++)
        printf("\t%s %d %s\t%s\n",
                scr[INV], i+1, scr[NOR], batchlist[i]);
printf("\n\n\tWelches Batch soll ausgefuehrt werden
        ('0' = Abbruch)? ");
scanf("%d", &which);

if(which > anz || which < 0)
        goto wahl;

if(which == 0)
        exit();

puts(scr[CLS]);

strcat(pfad, batchlist[which-1]);
sprintf(command, "Execute %s", batchlist[which-1]);
Execute(command, 0L, 0L);
}
```

Listing 1: WitchBatch ermöglicht die bequeme Auswahl mehrerer Batch-Files per Tastendruck.

"Witchbatch" in C erstellt und kann sowohl unter Lattice als auch unter Aztec compiliert werden. Wenn Sie im Besitz des Lattice-Compilers sind, geben Sie bitte folgendes ein:

```
lc -L Witchbatch
```

Daraufhin wird das Programm compiliert und gelinkt und kann danach aufgerufen werden.
Für Aztec Besitzer gilt folgendes:

```
cc Witchbatch
```

für die Compilation,

```
ln Witchbatch.o -lc
```

für das Linken der Objekt-Datei.

VON GERO PROBST

BASIC GURU

Alarmmeldung in AMIGABASIC



Das Programm BASICGURU erzeugt einen Alert, wie er z.B. als "GURU MEDITATION" bekannt ist. In Basic ist er recht kurz und einfacher aufgebaut als in C oder Assembler.

Zum Programm:

Eine Voraussetzung für die Funktionsfähigkeit des Programms ist das Vorhandensein der INTUITION-Bibliothek in einem Ordner LIBS auf der Diskette (natürlich können Sie den Suchpfad der Bibliothek auch ändern!).

Am Anfang wird die "DisplayAlert"-Routine aus der Intuition-Bibliothek geladen und die Alertnummer auf 1 gesetzt. Dies bedeutet, daß es sich um einen "Recoverable Alert", also einen Alarm, der nicht zum Systemabsturz führt, handelt. Das Gegenteil ist der "Dead End"-Alert, bei dem man nur noch hoffen kann, daß schon alles abgespeichert ist.

Der Bereich "Schrift" stellt den Text der Meldung zusammen. Dabei gehören zu jeder Zeile einige Steuersequenzen, die die Position des nachfolgenden Textes in der Meldung bestimmen. Abgeschlossen wird der Text-String mit CHR$(0)+"".

Danach erfolgt der Aufruf der Systemfunktion:

```
Bild& = DisplayAlert&(AlertNum$,
SADD(Text$),60)
```

Die letzte Zahl dieses Aufrufs gibt dabei die Höhe des Alarm-Kastens an. Nach diesem Aufruf erfolgt die Auswertung der erhaltenen Meldung, die

```
' ************************************************
' *  Programm zur Erzeugung von ALARM-Meldungen  *
' *                                              *
' *      von G. PROBST   Copyright KICKSTART     *
' *                                              *
' *  benoetigte Bibliothek: "Inuition.library"   *
' *                                              *
' ************************************************

Start:
  DECLARE FUNCTION DisplayAlert& LIBRARY
  LIBRARY "libs/intuition.library"
  alertnum&=1

Schrift:
  s1$=CHR$(0)+CHR$(130)+CHR$(20)
  t1$="Dieses Programm erzeugt einen 'RECOVERABLE ALERT'"
  s2$=CHR$(0)+CHR$(1)+CHR$(0)+CHR$(80)
  t2$="-LINKS KNOPF- Listing   -RECHTER KNOPF- Ende !"
  s3$=CHR$(0)+CHR$(1)+CHR$(30)+CHR$(100)
  t3$="von G.PROBST  88 fuer KICKSTART)"
  text$=s1$+t1$+s2$+t2$+s3$+t3$+CHR$(0)+""

Bearbeitung:
  Antwort&=DisplayAlert&(alertnum&,SADD(text$),60)
  IF Antwort&=1 THEN  ' linke Maustaste
     CLS:LIST
   ELSE               ' rechte Maustaste
     SYSTEM
   END IF
```

Listing 1: So leicht kann der GURU kommen - doch mit Absicht versteht sich.

ENDE

KICKS FÜR

INSIDER

In eigener Sache

Liebe Leser, KICKS für INSIDER ist eine Rubrik, die für interessante und lehrreiche Programme geschaffen wurde, denn was nutzt das schönste Programm, wenn man es nur abtippen, aber nicht verstehen kann.

Um jeden Monat neue KICKS zu veröffentlichen, sind wir natürlich auch von unseren Lesern abhängig, denn irgendwann einmal sind die Einfälle jedes Programmierers erschöpft. Also, nehmen Sie an dieser Rubrik aktiv teil, ergänzen Sie sie mit Ihren eigenen Programmen. Werden Sie Autor in der KICKSTART.

Dabei sind alle auf dem AMIGA existierenden Programmiersprachen gewünscht. Egal, ob Basic, C, Assembler, Modula 2, Pascal oder andere.

Wie das geht?

Ganz einfach. Schicken Sie uns das Programm, samt dem zugehörigen Artikel, eventuellen Bildern und Hardcopies (nach Möglichkeit "gegrabbt") auf Diskette zu. Wir werden uns dann bei Ihnen melden. Ein interessantes Honorar wartet bei Veröffentlichung natürlich auch auf Sie.

Wenn Sie Fragen haben, rufen Sie doch einfach bei uns an.

Tel. 06196/481811
Mo-Fr. 11 bis 17 Uhr

Ihre KICKSTART-Redaktion

LESERFORUM

Briefe an KICKSTART

*Liebe Leser,
die uns erreichende Flut von Telefonaten ist beim besten Willen nicht mehr zu schaffen. Wir sind daher gezwungen, feste Zeiten anzugeben, zu denen unsere Redakteure für Sie zur Verfügung stehen. Sie können an folgenden Tagen Fragen an Redaktionsmitglieder stellen: Dienstags von 14-17 Uhr
und Donnerstag von 14-17 Uhr
Bitte haben Sie dafür Verständnis, daß wir an anderen Tagen keine Fragen zu Soft- oder Hardware-Problemen entgegennehmen können.*

Sehr geehrte Damen und Herren,
seit ich meinen Amiga 1000 vor gut einem Jahr kaufte, bin ich ein großer Fan dieses Computers. Ich war dann auch glücklich, als eine spezielle Zeitschrift hierfür erschien, die ich nun seit dem ersten Heft verfolge. Die Informationsvielfalt dieses Magazins ist gut ausgewählt, so daß für jeden hier etwas zu finden ist.
Seit einigen Wochen bin ich nun auch Besitzer eines Amiga 2000 und habe hier einige Startschwierigkeiten. Zu meinen am häufigsten gebrauchten Anwendungen gehörte bisher der Flightsimulator II, der auf dem A1000 problemlos lief. Ich besitze dieses Programm mit den für den Amiga verfügbaren Scenerydisks. Auf dem A2000 nun bereitet mir die Arbeit mit diesem Programm einige Sorgen. Auch ein softwaremäßiges Abschalten des 1 MB-Speichers bringt hier keine Abhilfe, nach dem Start sind bei der Betätigung irgendeiner Taste alle Programmfunktionen tot. Kann mir hier jemand weiter helfen? Tritt dieses Problem auch anderweitig auf? Wie muß man gegebenenfalls die Workbench modifizieren, um Abhilfe zu schaffen?
Wolfgang Gillißen /Viersen

Leider tritt das von Ihnen beschriebene Problem häufiger auf. Die Ursache der aufgeführten Symptome liegt in der Tastatur begründet. Commodore hat in der Schaltungsauslegung der Tastatur einige kleine Änderungen vorgenommen, so daß die Tastatur des A2000 in ihrem Verhalten nicht mehr hundertprozentig mit einer A1000-Tastatur übereinstimmt. Viele Programme, die den Tastatur-Port direkt abfragen, kommen so in Schwierigkeiten. Da die auf dem Markt befindlichen Tastaturserien jedoch nicht alle vom selben Hersteller stammen, könnte ein Austauch der Tastatur Ihre Probleme beheben. Eine softwaremäßige Lösung oder eine Schaltung, die diesen Fehler beheben könnte, liegen uns leider nicht vor. Die Red.

Sehr geehrte Damen und Herren,
als Eigentümer eines Amiga 1000 mache ich von Zeit zu Zeit auch Gebrauch von der softwaremäßigen Emulation eines IBM-PCs, um Standardsoftware benutzen zu können. Da die Emulation allerdings an Geschwindigkeit zu wünschen läßt, interessiere ich mich für das von Ihnen getestete Hurricane Board mit 68020 und 68881.
Natürlich soll die höhere Leistung auch für normale Amiga-Programme genutzt werden.
Meine Frage speziell aber ist:
Verträgt das Board auch die Softwareemulation bzw. arbeitet das Board auch mit diesem Programm so geschwindigkeitssteigernd zusammen?
Peter Witting/Trier

Der IBM-Emulator verträgt sich nicht ohne weiteres mit einem 68010 oder 68020. Um jedoch diese Programm auch mit dem Hurricane-Board nutzen zu können, wird von der Herstellerfirma dieser Karte ein Patchprogramm mitgeliefert, welches den Betrieb der Emulationssoftware in Verbindung mit einem 68020 ermöglicht. Die Geschwindigkeitssteigerung entspricht den im Test erwähnten Zeiten. Da jedoch einige Versionen des Emulators auf dem Markt sind, können wir hier keine hundertprozentige Aussage treffen. Sie müßten sich hierzu an einen Fachhändler wenden, der diese Karte führt, und dort eventuell Ihre Software testen. Die Red.

Liebe Redaktion,
ich bin Besitzer eines Amiga 500 und wollte meinen Drucker, einen NEC P7, den ich schon an einem PC ohne Probleme angeschlossen hatte, auch an der obigen Maschine ausprobieren, doch er streikt, oder druckt vielmehr keine richtige Grafiken. Ich habe den Drukker als CBM MPS 1000 angemeldet, er steht auf 8 lpi, und trotzdem gibt es das mitgeschickte Ergebniss. Wo liegt mein Fehler?

Der NEC P7 ist baugleich mit dem Commodore Drucker MPS 2000. Sie müssen daher auch diesen Treiber einstellen. Dann sollte sich auch ein entspechender Grafikausdruck realisieren lassen. Wenn dieser Treiber nicht auf Ihrer Workbench vorhanden ist, können Sie ihn bei Ihrem Händler bekommen. Auf älteren Workbenchs war dieser nämlich nicht enthalten.

Die Red.



ABO





ABO

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte mit 60 Pf. frankieren

Heim Verlag
Heidelberger Landstr. 194

6100 Darmstadt-Eberstadt
Telefon 0 61 51 / 5 60 57



Einzelheft- u. Monatsdisketten Bestellung





Einzelheft- u.
Disketten Service

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte mit 60 Pf. frankieren

Heim Verlag
Heidelberger Landstr. 194

6100 Darmstadt-Eberstadt
Telefon 0 61 51 / 5 60 57



Kontaktkarte





Kontaktkarte

Bitte Adresse der Firma, bei der Sie Informationen, oder etwas bestellen möchten, auf der rechten Seite eintragen → → → → → → → → → →

Absender
(Bitte deutlich schreiben)

Vorname/Name

Beruf

Straße/Nr.

PLZ/Ort

Telefon Vorwahl/Rufnummer

Postkarte

Bitte freimachen

Firma

Straße/Postfach

PLZ Ort

KICK START AMIGA ZEITSCHRIFT

Abonnement

Ja, bitte senden Sie mir die Amiga-Computer Fachzeitschrift ab ______
für mindestens 1 Jahr (11 Hefte) zum ermäßigsten Preis von jährlich DM 70,— frei Haus.
(Ausland: Nur gegen Scheck-Voreinsendung DM 90,— Normalpost.)
Der Bezugszeitraum verlängert sich nur dann um ein Jahr, wenn nicht 6 Wochen vor Ablauf des Abonnements gekündigt wird.

Name ______

Vorname ______

Straße/Nr. ______

PLZ ______ Ort ______

Gewünschte Zahlungsweise bitte ankreuzen

☐ Bequem und bargeldlos durch Bankeinzug

Konto-Nr. ______ BLZ ______

Institut ______ Ort ______

☐ Ein Verrechnungsscheck über DM______ liegt bei.

Garantie:
Diese Bestellung kann ich schriftlich innerhalb einer Woche (rechtzeitige Absendung genügt) widerrufen. Dies bestätige ich durch meine 2. Unterschrift.

Datum ______ Unterschrift ______

Datum ______ Unterschrift ______

KICK START AMIGA ZEITSCHRIFT

ABO

KICK START AMIGA ZEITSCHRIFT

Einzelheft- u. Monatsdisketten

Bestellung

KICKSTART können Sie direkt beim HEIM-VERLAG zum Einzelheft-Preis von DM 7,– (zuzüglich Gebühr für Porto und Verpackung) nachbestellen. Bearbeitung nur gegen beigefügten Scheck über den entsprechenden Betrag (keine Überweisung).

Jan.	Febr.	März	April	Mai	Juni	Juli/Aug.	Sept.	Okt.	Nov.	Dez.	
											1987 = DM
											1988 = DM

+ Gebühr für Porto u. Verp. = DM

☐ Scheck in Höhe zus. DM liegt bei

Disketten Service

Alle Programme, die in KICKSTART veröffentlicht wurden, sind auf Disketten erhältlich. Die Disketten enthalten die Programme von jeweils 2 KICKSTART-Ausgaben. Bestellen Sie durch ankreuzen die gewünschten Disketten

Preis je Diskette 19,– DM	Juli/Aug.	Sept./Okt.	Nov./Dez.	Jan./Febr.	März/Apr.
	87	87	87	88	88

Lieferung: gegen beigefügten Scheck zuzügl. 5,– DM Versandkosten

KICK START AMIGA ZEITSCHRIFT

Einzelheft- u. Monatsdisketten Bestellung

KICK START AMIGA ZEITSCHRIFT

☐ Ich bitte um weitere Informationen

☐ Ich gebe folgende Bestellung auf

in Bezug auf Ihre Anzeige in Kickstart Heft ______ Seite ______



Menge	Produkt/Bestellnummer	DM	gesamt DM

Datum, Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

Abgesandt am:

KICK START AMIGA ZEITSCHRIFT

Firma:

Bemerkungen:



Kontaktkarte





Kontaktkarte

Bitte Adresse der Firma, bei der Sie Informationen, oder etwas bestellen möchten, auf der rechten Seite eintragen → → → → → → → → → → →

Absender
(Bitte deutlich schreiben)

Vorname/Name

Beruf

Straße/Nr.

PLZ/Ort

Telefon Vorwahl/Rufnummer

Postkarte

Bitte freimachen

Firma

Straße/Postfach

PLZ Ort



Kurzmitteilung





Kurzmitteilung

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Telefon

Postkarte

Bitte freimachen

MERLIN Computer GmbH
KICKSTART Redaktion
Industriestraße 26

6236 Eschborn



PD Bestellung





PD Bestellung

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte freimachen

MERLIN Computer GmbH
KICKSTART Redaktion
Industriestraße 26

6236 Eschborn



☐ Ich bitte um weitere Informationen

☐ Ich gebe folgende Bestellung auf

in Bezug auf Ihre Anzeige in Kickstart Heft ________ Seite ____

Menge	Produkt/Bestellnummer	DM	gesamt DM

Datum, Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)



Abgesandt am:



Firma:

Bemerkungen:



Meine Meinung

Zu dem Artikel ____________ in Heft ____, Seite ____
hätte ich folgendes zu bemerken:

☐ Ich möchte Ihnen folgendes Programm zur Veröffentlichung anbieten: (Kurzbeschreibung, Sprache, Länge in Druckerseiten)

☐ Ich kann über folgendes Thema berichten: (Tips & Tricks am AMIGA, Hardware, Software, etc.)

☐ Ich möchte gerne Autor in der KICKSTART werden. Meine Fachgebiete: (z.B. LISP, Modula-2, Assembler...)

☐ Ich möchte, daß folgendes Public-Domain Programm in Ihre Sammlung aufgenommen wird.

☐ Sonstiges

Bei weiteren Angaben oder Fragen wenden Sie sich bitte schriftlich oder telefonisch an die Redaktion. Tel. 0 61 96/48 18 11





Kurzmitteilung



PUBLIC DOMAIN SERVICE

Ich bestelle folgende PD-Disketten:
(Siehe PD Service in dieser Ausgabe)

Zahlung erfolgt:

☐ per Scheck

☐ per Nachnahme

Je Diskette fügen Sie bitte einen Betrag von DM 10,– bei,
für Porto und Verpackung je Sendung DM 5,– (Ausland DM 10,–)

Datum

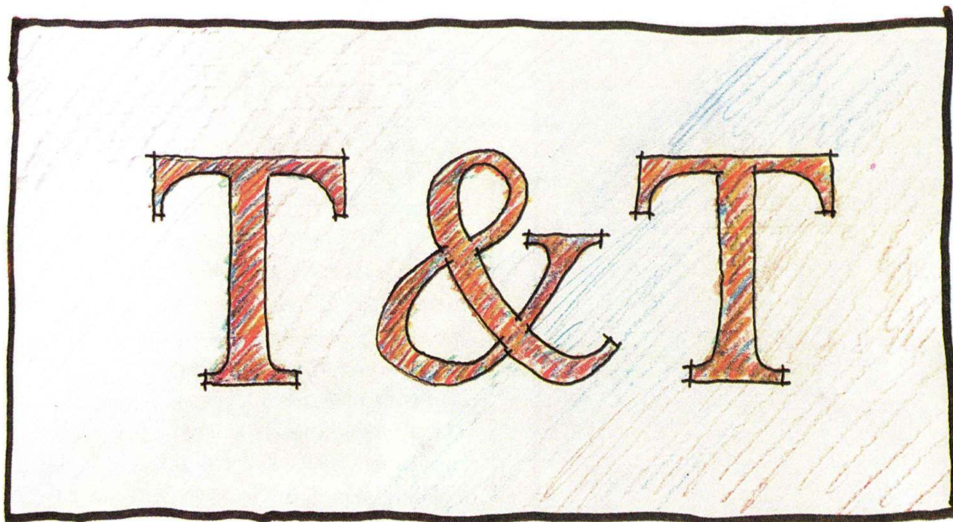
Unterschrift





PD Bestellung

Datum



FULLSCREEN auch in AMIGABASIC

In der Ausgabe 2/88 der KICKSTART wurde der Artikel "FULLSCREEN" für den PAL-AMIGA veröffentlicht. Da das Programm jedoch in C geschrieben war, konnten es viele interessierte Leser nicht verwenden. Michaela Nie-Hoegen hat deshalb das Programm in AmigaBasic umgeschrieben.

Die Erläuterung des Vorgangs kann dem Artikel in Heft 2/88 entnommen werden. Das Programm öffnet die Datei "SYSTEM-CONFIGURATION" in Laufwerk DF0: und schreibt an die Positionen 119,120,217 und 218 die neuen Werte für den Workbench-Bildschirm.

Dazu wurde ein Unterprogramm CHANGE verwendet, dem die Parameter Position, Wert und die 1-Byte-Variable t$ übergeben werden. Das Unterprogramm gibt dann den Wert aus, der vorher in der Datei stand, was nur zur Kontrolle dient.

Wenn das Programm fertig ist, muß man den Rechner erneut Booten und erhält danach einen vergrößerten Workbench-Bildschirm.

Anmerkung: Bevor man das Programm startet, sollte man eine Kopie der Workbench-Diskette anfertigen, damit man vor dem Fall der Fälle gesichert ist.

(Michaela Nie-Hoegen, Loffenau)

```
REM ****************************************
REM *      FULLSCREEN   672x274 Punkte      *
REM *                                       *
REM *      Version in AmigaBasic von        *
REM *         Michaela Nie-Hoegen           *
REM ****************************************

PRINT "SYSTEM-CONFIGURATION wird geändert !"

OPEN "r",#1,"df0:devs/system-configuration",1
FIELD #1,1 AS t$

CALL change(119,248,t$)
CALL change(120,246,t$)
CALL change(217,18,t$)
CALL change(218,32,t$)

CLOSE #1
PRINT "Aktivieren durch neues Booten der WB !"
END

SUB change(code%,wert%,t$) STATIC
   GET #1,code%
   PRINT "alter Wert: ";ASC(t$);: INPUT a$
   LSET t$=CHR$(wert%)
   PUT #1,code%
END SUB
```

KISSING BYTE BANDIT GOOD-BYE !!!

Als AMIGA-500-Neuling war ich über die Infektion mit dem "Byte Bandit-Virus" zuerst entsetzt, weil ich nicht viel über ihn wußte. Desto mehr freute ich mich über den Artikel in der KICKSTART 4/88.

Wer sich den Virus noch einmal anschauen möchte, der findet ihn im Bootblock (Cylinder 0, Sektor 0, Head 0, Block 0). Zwei Punkte sind zur Entdeckung und Vernichtung von "Byte Bandit" wichtig:

1. Der Virus ist resetfest.
2. Wenn das System nach dem Einschalten mit einer virusfreien Diskette gebootet wurde, dann kann sich der Virus nicht festsetzen, es sei denn, man führt einen Reset aus.

Die folgende Anleitung ist speziell für AMIGA-Fans gedacht, die nur ein Laufwerk besitzen. Wichtig ist dabei aber, daß die Diskette, mit der Sie booten, virusfrei ist, denn sonst hilft die ganze Arbeit nichts! Kontrollieren Sie deshalb am besten die in Frage kommende Diskette mit einem Diskettenmonitor (z.B. DISKZAP von KICKSTART PD 7). Danach schreiben Sie die folgende Batch-Datei, die unter dem Namen Virus.Killer abgespeichert wird. Der Aufruf erfolgt dann mit: EXECUTE Virus.Killer. Das Programm kopiert dann einige CLI-Befehle auf die Ram-Disk. Danach werden Sie aufgefordert, die infizierte Diskette einzulegen. Die Bootspur dieser Diskette wird nun neu geschrieben, und der Virus ist gebannt.

Achtung: Bei manchen Original-Disketten, die ihren Kopierschutz in der Bootspur abgelegt haben, bewirkt die Prozedur, daß diese Diskette nicht mehr als Original erkannt wird!

(Stefan)

Batch-Datei VIRUS.KILLER kann z.B. mit ed Virus.Killer eingegeben werden:

```
ECHO "richte RAMDISK ein, bitte warten ..."
COPY c/(cd|echo|install|wait) to ram:
CD ram:
ECHO "Nach Verlöschen der Laufwerksdiode legen"
ECHO "Sie bitte die infizierte Diskette ein !"
WAIT 10 SECS
INSTALL df0:
ECHO "Kissing BYTE BANDIT goodbye !!!"
ECHO "Bitte Bootdiskette wieder einlegen!"
WAIT 10 SECS
CD df0:
DELETE ram:#?
ECHO "Es ist geschafft ..."
```

Die Zeichensätze des NEC P2200

Leider ist es beim AMIGA nicht so ohne weiteres möglich, besondere Fähigkeiten eines Druckers anzusprechen. So ist es z.B., wie auf anderen Rechnern üblich, nicht möglich, mit LPRINT Escape-Sequenzen zur Einstellung von Schriftarten oder Zeichensätzen an den Drucker zu übermitteln. Dies liegt daran, daß LPRINT den Datenkanal "PRN:" benutzt, der die Bedeutung der Escape-Sequenzen aus dem Druckertreiber des AMIGA liest. Deshalb ist es auf diesem Weg nicht möglich, die verschiedenen Zeichensätze des NEC P2200 umzuschalten.

Eine Lösung dieses Problems bringt der Datenkanal "PAR:". Dieser wird von Basic aus als Datei geöffnet. Zwar müssen nun alle LPRINT-Anweisungen "PRINT #2" lauten, doch kann man jetzt die Escape-Sequenz "ESC k (n)" in Form von CHR$(27)+"k"+CHR$(font) an den Drucker übermitteln. Das untenstehende Beispielsprogramm übergibt dem Unterprogramm "ChangeFont" die zuvor definierten Werte der Zeichensätze. Danach wird das Unterprogramm "Test" aufgerufen, welches das Alphabet ausdruckt.

```
PRINT "Die FONTS des NEC P2000"

Courier=0
Focus=6
OCR=12
Souvenir=15

OPEN "par:" FOR OUTPUT AS #2
ChangeFont(Courier): CALL Test
ChangeFont(Focus):   CALL Test
ChangeFont(OCR):     CALL Test
ChangeFont(Souvenir): CALL Test

CLOSE 2
END

SUB ChangeFont(font) STATIC
  IF font=0 OR font=6 OR font=12 OR font=15 THEN
    PRINT #2,CHR$(27)+"k"+CHR$(font)
  END IF
END SUB

SUB Test STATIC
  PRINT #2,"abcdefghijklmnopqrstuvwxyz";
  PRINT #2,"ABCDEFGHIJKLMNOPQRSTUVWXYZ"
END SUB
```

TIP zu PINBALL WIZARD

Wenn man PINBALL WIZARD von KINGSOFT einige Zeit spielt, wird der Sound recht eintönig. Neuen Reiz kann man schaffen, indem man die Soundfiles des Programms manipuliert. Als erstes sollte man die Diskette jedoch kopieren und nur mit dieser Kopie arbeiten. Die Soundfiles befinden sich im Ordner SFX. Die einfachste Möglichkeit ist nun, die Namen der Files zu vertauschen, was zu sehr interessanten Effekten führen kann. Eine anderer Weg ist, andere oder selbst aufgenommene Soundfiles im Flipper einzusetzen.

Hierbei ist zu beachten, daß die Abtastrate 10.000 Hz betragen und der Sound mit DUMP abgespeichert werden sollte. Die Länge ist dabei im Prinzip variabel, sie sollte jedoch nicht zu stark von der ursprünglichen abweichen.

Durch diese Technik der Verfremdung wird PINBALL WIZARD deutlich abwechslungsreicher, außerdem kann sich jeder ein indiviuelle Geräuschkulisse zusammenstellen.

(Thorsten Gau, Neu-Wulmstorf)

NEUESTE

Vom Spielemarkt

ROLLING THUNDER

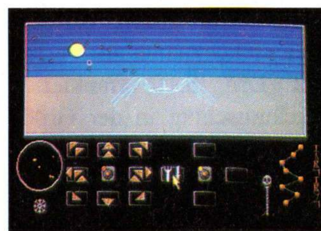
U.S. GOLD präsentiert mit "Rolling Thunder" eine weitere Automatenumsetzung für den Amiga. Als Agent einer Geheimorganisation kämpft man sich durch das Hauptquartier eines Gangsters, der die Weltherrschaft an sich reißen möchte. Das mittelprächtig animierte Heldensprite läuft von links nach rechts durch einen langen Gang, von dem mehrere Türen abzweigen. Zunächst ist er nur mit einer Pistole bewaffnet. Die Bewaffnung ändert sich allerdings im Verlauf des Spiels. Hinter den Türen kann sich der Held beispielsweise ein Maschinengewehr holen und kräftig unter den bösen Feinden "aufräumen". Der Held von "Rolling Thunder" ist ein komischer Zeitgenosse, grafisch hätte man etwas mehr Sorgfalt walten lassen sollen. Er springt mit angezogenen Beinen im Känguruhsprung über im Weg stehende Kisten. Wird man während eines Sprungs in der Luft getroffen, sackt der Superagent wie ein Mehlsack zusammen und bleibt in der Luft schweben. ROLLING THUNDER auf dem AMIGA muß nicht unbedingt sein.



Die Grafik erinnert etwas an den guten, alten C64.

GENMANIPULATION MIT OCEAN

ECO heißt eine gelungene Simulation vom englischen Softwaregiganten OCEAN, der mit ECO auch in den 16-Bit-Spielemarkt einsteigt. In ECO dreht sich alles um die sogenannte Genmanipulation. Mit der Maus steuert man eine primitive Spezies über einen Planeten. Ziel des Spiels ist es, aus dem primitiven Wesen eine hochentwickelte Spezies zu entwickeln. Dazu muß man mit seinem Primitivum, das wie eine Spinne aussieht, zuerst Nahrung aufnehmen. Es ist darauf zu achten, daß man dabei micht zum Opfer höher entwickelter Wesen wird. Wenn man sich sattgefuttert hat, muß man sich ein Weibchen suchen, mit dem man sich paart. Bei den ersten Nachkommen kann man daraufhin die Gene manipulieren. Ist dies geschehen, steuert man die neuentstandene Spezies über den Planeten, und der ganze Vorgang wiederholt sich solange, bis man zur Creme de la Creme der Schöpfung gehört. Alle Spielcharaktere präsentieren sich in einer ruckelfreien, gut animierten Vektorgrafik. Vor dem Kauf sollte man sich das Spiel einmal ansehen: Da man nur die Umrisse der dargestellten Gegenstände oder Personen sieht, ist Vektorengrafik sicherlich nicht jedermanns Sache.



Die Vektorgrafik bietet leider keine ausgefüllten Flächen.

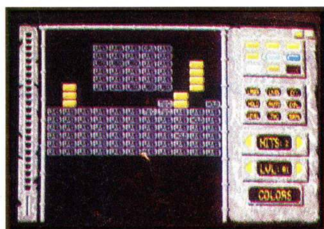
DREIFACHE SCHALLGESCHWINDIGKEIT

Mit einer klar verständlichen Frauenstimme, die euphorisch "Get Ready for Mach 3" schreit, wird der Spieler von MACH 3, dem ersten Amigaspiel des französischen Softwarehauses LORICIELS, begrüßt. Grafisch ist MACH 3 ganz ordentlich geworden. Das Raumschiff, das der Spieler über eine Planetenoberfläche steuert, ist exellent animiert. Doch im Spiel kommt nur Langeweile auf. Minutenlang passiert überhaupt nichts. Dann kommen plötzlich ein paar feindliche Jagdraumschiffe, die man abschießen muß, und dann geht die Langeweile wieder von vorne los. Sound und Grafik sind OK. Der Rest - Nein danke!

BREAKOUT - DIE 99'ZIGSTE

CRACK, das neue Spiel des Schweizer Softwarenewcomers LINEL, ist, wie kann es anders sein, die obligatorische Breakoutversion für diesen Monat. Beim Betrachten der Titelsequenz läuft dem Spielefeinschmecker das Wasser im Munde zusammen. Drei Steinzeitmenschen meißeln in herrlicher Bilderbuchgrafik den Spieletitel in

einen Felsen. Grafisch ist diese Sequenz sehr eindrucksvoll gestaltet. Auch der Rest des Spiels hat eine sehr ansprechende Grafik zu bieten. Über die Soundeffekte kann man sich ebenfalls nicht beklagen. Programmierte Musik mit digitalisierten Effekten und die digitalisierten Urwaldlaute sind vom Feinsten, was man auf dem Amiga bisher hören kann. Exellente Grafiken, der gut gelungene Sound, 150 Level, ein Zwei-Spieler-Simultanspielmodus und ein komfortabler Screeneditor können aber nicht darüber hinwegtäuschen, daß es sich bei Crack um den 99zigsten Aufguß eines uralten Spielprinzips handelt.



Ein Editor ist bei CRACK integriert.

ORIGINALE FÜR 9.99 DM

Das Gerücht war schon seit einiger Zeit im Umlauf, jetzt hat CACHET als erste Softwarefirma für den Amiga den absoluten Niedrigpreis von 9.99 DM pro Spiel wahrgemacht. Galt bisher die Billigspielserie von Kingsoft mit 29.- DM als nicht zu unterbieten, so ist der Preis von CACHET eigentlich kaum zu glauben.

Nach den Staunen stellten sich dann sofort auch zahlreiche Fragen: Was erhält der Kunde für diesen Preis? Kann man hier von "Qualität" in entferntesten Sinn noch sprechen? Und wie will CACHET dabei noch zu Gewinn kommen? Die Antworten haben uns in der Tat überrascht! Der Kunde erhält ein komplettes Spiel mit marktüblicher Verpackung.

Die Qualität der Spiele ist nicht etwa schlecht, sondern im Gegenteil durchaus ansprechend. Besonders grafisch waren die Spiele besser als manches, was sich in der Billigspielklasse tummelt.

Die Frage nach dem Gewinn hat uns CACHET auch beantwortet: keine Werbung und die Hoffnung auf sehr hohe Verkaufszahlen aufgrund des geringen Preises lassen bei einer Gewinnspanne von 1.- DM pro Spiel die Chance auf das Gelingen dieses Experiments möglich erscheinen. Dazu gehört auch, daß die Amigafreaks sich diese Spiele kaufen, statt sie mit dem üblichen Argument "zu teuer" nur raubkopieren. CACHET hat konsequenterweise auf einen Kopierschutz verzichtet und statt dessen den Slogan geprägt: "Unsere Spiele sind so günstig, daß sich raubkopieren nicht lohnt."

Einen Haken hat die Software für 9.99 DM aber doch: Um diesen Preis realisieren zu können, ist es nicht möglich, die Software über Händler zu vertreiben. Alle Programme können nur direkt bestellt werden. Die enge Kalkulation läßt einen Händlerrabatt nicht mehr zu.

Wer das erste Spiel "Revenge of Doom" bestellen möchte, muß neben dem Zehnmarkschein oder einem Verrechnungsscheck auch noch einen mit DM 3.10 frankierten Rückumschlag in der Größe DIN A4 beilegen oder eine Versandkostenpauschale von DM 4.- entrichten.

Die Bezugsadresse:
CACHET, Ostendstr. 32, 7524 Östringen

ELECTRONIC ARTS

"Arts", wie EA von Freunden des Hauses kurz und liebevoll genannt wird, hat in letzter Zeit als #1 in der Publikumsgunst Federn lassen müssen. Grund dafür waren unter anderem die als "Endversionen" verbreiteten Raubkopien der Vorabversionen von "Return to Atlantis" und "Ferrari Formula One". Mittlerweile sind aber die tatsächlichen Endversionen erschienen, und "Arts" erhebt sich wie der Phönix aus der Asche. "Ferrari Formula One" ist von allen bisher erschienen Autorennspielen das, das den Titel "Rennsimulation" am ehesten verdient. Zahlreiche Optionen und ein sehr komplexer Aufbau sind ohne Anleitung kaum zu durchschauen. Genau wie bei "Return to Atlantis" (siehe Spieleteil) schlägt EA damit den Softwarepiraten ein Schnippchen. Die illegale Kopie wird dadurch zur "Demoversion" degradiert.



EA präsentiert eine Formel 1-Simulation mit allen Schikanen.

LIEBESHEIRAT ODER VERNUNFTEHE?

Hat hier nun eine "heimliche Hochzeit" stattgefunden oder war das eher eine Adoption? Und wenn ja, wer hat hier wen ...?

Welchem glücklichen Umstand wir Amigabesitzer die "Verbindung" zwischen ACTIVISION und MICROILLUSIONS verdanken, wollte man uns nicht genau erklären. Immerhin, wir haben herausgefunden, daß die "zarten Bande" zwischen den beiden Softwaregiganten erst einmal nur für den "deutschen Sprachraum" (ohne Weißrußland?!) gelten. Resultat: ACTIVISION ist ab sofort EXKLUSIVVERTRETER aller Produkte von MI für das oben angeführte Gebiet.

Als "Hochzeitsgeschenk" an uns werden deshalb die Preise der MI-Produkte für den deutschen Markt erst einmal kräftig gesenkt (z.B.: Faery Tale Adventure 89.-DM).

Als Geschenk des werten Publikums wünscht man sich eine, durch die gesenkten Preise zusätzlich angestachelte, gesteigerte Nachfrage.

Die Antwort darauf sollen unter anderem EBONSTAR und der zweite Teil von FAERY TALE ADVENTURE sein.

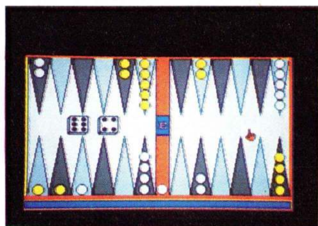
Während das Letztere leider erst im Spätsommer erscheinen wird, hat das Erste soeben das Licht der Softwareshops erblickt.

Bei EBONSTAR handelt es sich um die Amiga-Variante eines bekannten Arcade-Games. Auf einer Weltraummatrix wird ein alles verschlingendes schwarzes Loch von einem gegnerischen Raumschiff mittels eines Traktorstrahls herumgeschleppt. Die "Guten" und die "Bösen" (bzw. die Grünen und die Roten) versuchen sich nun gegenseitig mit gezielten Schüssen in dieses schwarze Loch zu treiben. Die Spielidee ist schlicht, aber faszinierend. Diese Eigenschaften teilen sich auch Grafik und Musik - und zwar genau in der aufgeführten Reihenfolge.

CHAMPIONSHIP BACKGAMMON

"...Ein Vorläufer dieses Spiels wurde bereits vor Jahrtausenden gespielt. Mit der Zeit hat es sich zu einem aufregenden Spiel entwickelt, bei dem Zufall und Taktik

Hand in Hand gehen...".
Anhand dieses Zitats aus der Anleitung ist klar ersichtlich, daß die Werbestrategen von Robtek sich mit der Geschichte des Backgammon ein wenig auskennen. Den Programmierern dieses neuesten ABasic-Hits (kein Scherz!!!) kann man allenfalls bescheinigen, daß sie sich noch in der Programmiersteinzeit befinden. Bereits im zweiten Zug (ein Dreier-Pasch) hat Championship Backgammon 1:27 Minuten gebraucht, um sich zu entscheiden. Trotzdem war die Spielstärke so schwach - oder besser, die Spielschwäche so stark, daß ein mäßiger Spieler durchaus gewinnen kann. "King Size" - das Label - deutet hier wohl eher auf die Größe des Bettes hin, in dem der Programmierer sein Geld im Schlaf zu verdienen hofft.



Auch die Grafik ist eher als müde zu bezeichnen.

TIME AND MAGIC

Unter diesem Titel sind die drei Abenteuer LORDS OF TIME, RED MOON und THE PRICE OF MAGIK als Trilogie in einem Paket zusammengefaßt. Sie wurden bereits an anderer Stelle der KICKSTART besprochen. Das Komplettpaket vermarktet LEVEL 9 zusammen mit MANDARIN SOFTWARE.

BUBBLE BOBBLE

Der lustige Spielhallenhit von TAITO wurde jetzt von Firebird umgesetzt. Bub und Bob, die kleinen Drachen, machen sich auf den Weg durch 100 Level voller Gefahren, um ihre Herzallerliebsten zu finden. "Bewaffnet" mit Seifenblasen trotzen sie allen Gefahren. Trotz der etwas "einfachen" Grafik ist dieses Spiel ganz sicher einer der kommenden Verkaufshits.



Bubble Bobble besticht nicht durch gute Grafik, sondern eher mit Motivation.

VOM AMIGA IN DIE SPIELHALLE

Direkt vom Amiga in die Spielhalle portiert werden möglicherweise mehrere der zur Zeit in Entwicklung befindlichen Spiele von Discovery Software. Neben ZOOM (siehe Spieleteil) sind vor allem HYBRIS und ein Spiel mit dem Arbeitstitel SODAN - DER UNBESIEGBARE dafür bestens geeignet. Während ZOOM soeben erschienen ist, wird der Amiga-Spielefreak noch ca. zwei Monate warten müßen, bis er die beiden anderen Actiongames ins Laufwerk schieben kann. Spielbare Vorabversionen stehen uns allerdings bereits jetzt zur Verfügung.

SLAYGON

Der SLAYGON ist der bestentwickelte Kampfroboter aller Zeiten. Deshalb wird er ausgeschickt, um das kybernetische Labor der gegnerischen Streitkräfte zu zerstören.
Neben der Kampfstärke einer ganzen Hundertschaft besitzt er aber nur die Intelligenz seines Benutzers. Glücklicherweise gibt es in den verschiedenen Ebenen des Gebäudes mehr als drei Dutzend Objekte zu finden, die im entscheidenden Moment von unschätzbarer Hilfe sind. Vielleicht ist es nicht die "...most important mission in the history of humanity...", wie die Packung verspricht, aber Spaß macht die Mischung aus Strategie und Action ganz sicher.

FORMULA 1

Fünf verschiedene Rennwagen stehen dem Spieler von FORMULA 1, MICRO VALUES neuestem Spiel für den Amiga, zur Verfügung. Mit Ferrari, McLaren, Benetton, Lotus oder Williams begibt man sich auf die Rennpiste. Sechs Kurse und die Gangschaltung können nicht überzeugen.



Die 3D-Perspektive täuscht etwas über die Qualität hinweg.

Info: Ariolasoft, Leisuresoft, Rushware

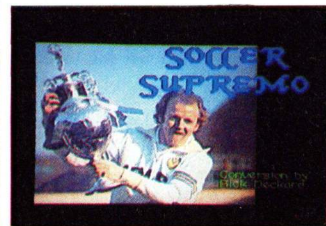
AUF GROSSER ENTDECKUNG

In die Zeit der Entdecker und Eroberer versetzt Sie THE SEVEN CITIES OF GOLD - Kolumbus läßt grüßen. Die gelungene Mischung aus Aktion- und Strategiespiel ist jetzt offiziell in Deutschland erhältlich und an die Betriebssystem-Version 1.2 angepaßt. Es gilt, einen fremden Kontinent zu erforschen und sich die Einheimischen zunutze machen. Ist ein Kontinent erforscht, kann man sich vom Computer einen neuen errechnen lassen.
Grafisch kann das Spiel von ELECTRONIC ARTS nicht überzeugen. Dafür stehen dem Spieler aber viele Spielfeatures zur Verfügung.
Im Hafen werden die Entdekkungsschiffe ausgerüstet. Man kann dann damit über die Ozeane schippern, Inseln anlaufen und mit den Eingeborenen Handel treiben. Wenn man auf grafische Elemente keinen gesteigerten Wert legt, kann man sich mit THE SEVEN CITIES OF GOLD gut und lang unterhalten.

Info: Rushware

SOCCER SUPPREMO...

...nennt sich ein neuer Aufguß des legendären Fußballmanagers. Hersteller CRUSUS hat ein gutes Fußballmanagerspiel geschaffen. Dummerweise hat man sich bei der Grafik während des Spiels nicht sonderlich viel Mühe gegeben. Die Menüs sind auf dem schlichten Blau-Untergrund zu sehen, den wir den wir von der Workbench kennen. Ab und zu sind einige digitalisierte Bilder zu bewundern.



Digitalisierte Bilder tauchen hin und wieder auf.

Info: Leisuresoft

TOP 12

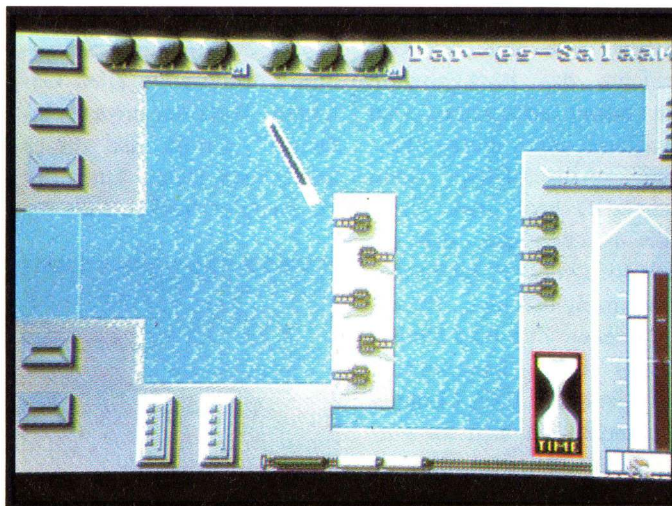
Auch in diesem Monat ist die Handelssimulation PORTS OF CALL wieder auf Platz 1 gewählt worden. Die Qualität des Spiels steht so ziemlich außer Zweifel, denn der Vorsprung ist sehr beachtlich. Auf Platz 2 bis 7 folgen die bewährten 'Klassiker' ARKANOID, FLIGHT II, EMERALD MINE, BARD'S TALE, BARBARIAN und SHANGHAI, die sich immer weiter nach oben schieben. Die nächsten drei Plätze werden von vielversprechenden Newcomern belegt. PINK PANTHER wurde im letzten Heft ausführlich vorgestellt, THE GREAT GIANA SISTERS und XENON schafften es auch ohne große Besprechung sich zu plazieren.

Mitmachen lohnt sich, denn unter allen Teilnehmern verlosen wir den TOP 12-TIP des Monats. Dies wird in der Regel ein brandneues Spiel sein, das bei der Redaktion besondern Gefallen gefunden hat. Lassen Sie sich also überraschen!

Mitmachen kann jeder, der die TOP 12-Postkarte (bitte keine andere!) ausfüllt und uns zuschickt. Der Rechtsweg ist dabei ausgeschlossen. Einsendeschluß ist der 30. Mai 1988. Karten, die nach diesem Termin eintreffen werden im nächsten Monat berücksichtigt.

Wir gratulieren den Gewinnern dieses Monats, die jeweils das Spiel 'ZOOM' erhalten.

1. (1) Ports of Call



2. (4) Arkanoid

3. (2) Flightsimulator II

4. (7) Emerald Mine

5. (8) The Bard's Tale

6. (11) Barbarian

7. (5) Shanghai

8. (-) The Great Giana Sisters

9. (-) Pink Panther

10.(-) Xenon

11.(9) Test Drive

12.(10) Defender of the Crown

Stefan Mefert, Osnabrück/ Boon Elschenbroich, Landsberg/ Carsten Sledz, Hannover/ B. Girschick, Günzburg/ Holger Himmelmann, Leimen/ Daniel Kölligan, Köln/ Peter Edl, Wolfratshausen/ Detlef Krischak, Bodenteich/ Michael Mertel, Würzburg/ Christian Schmalzgruber, Braunau/ Andreas Pramberger, Painten/ Sven Rowold, Frankfurt

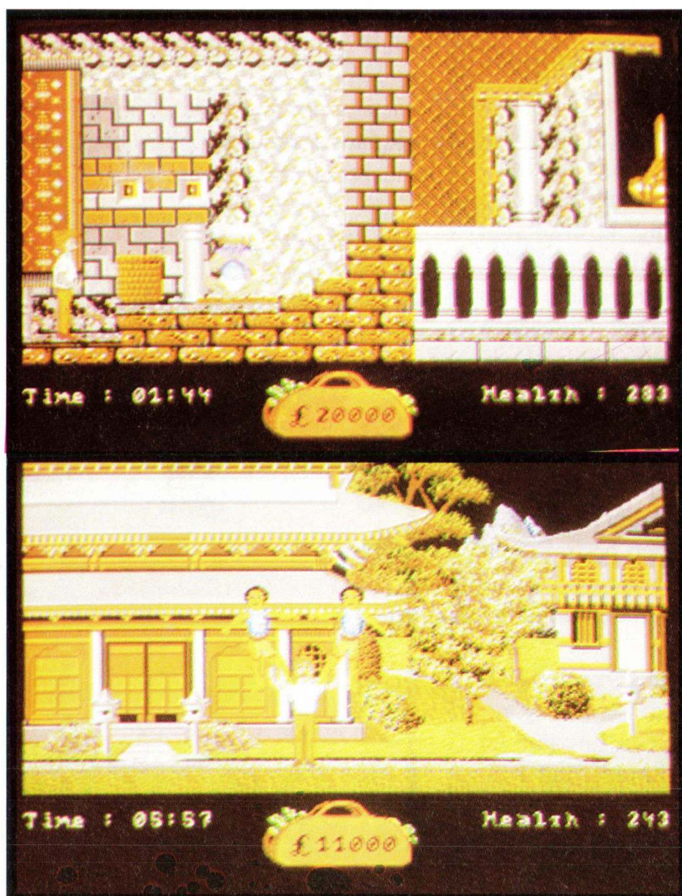
IN 80 TAGEN UM DIE WELT

Die Handlung orientiert sich an dem gleichnamigen Roman von Jules Verne. Protagonist Phileas Fogg hat sich auf eine verrückte Wette eingelassen. Geld hat der reiche Brite genug, darauf deutet der hohe Wetteinsatz von 20000 Pfund Sterling hin. Mit Hilfe dieser Geldsumme soll Fogg um die Welt reisen. Kein leichtes Unterfangen, zumal die Preise für Transportmittel recht hoch sind.

Nach einem gelungenen Vorspann mit exzellentem Stereosound kann man mit dem Abenteuerspiel beginnen. Es erscheint eine Weltkarte auf dem Bildschirm. Mit einem Schiff tuckert man über die Ozeane. Das geschieht natürlich unter enormen Zeitdruck, man hat schließlich nur 80 Tage Zeit, um die ganze Welt zu umreisen. Der Reiz von Rainbow Arts neuestem Spiel liegt aber nicht in der Weltkartenszene. Sechs verschiedene Actionszenen sorgen für hohe Spielmotivation. Jeder Actionteil ist völlig anders gestaltet. Im ersten Level befindet sich Fogg in Indien. Bewaffnet mit einem Speer, führt ihn das Abenteuer durch die Wildnis und durch einen geheimnisvollen Tempel. In Japan soll Fogg eine menschliche Pyramide bauen. Mit einem Eskimoschlitten geht das Abenteuer in Amerika weiter. Danach ist Fogg in eine wilde Schießerei verwikkelt. Alle Actionsequenzen präsentieren sich in einer ansprechenden Grafik, in der es von Farben und tollen Animationen nur so wimmelt.

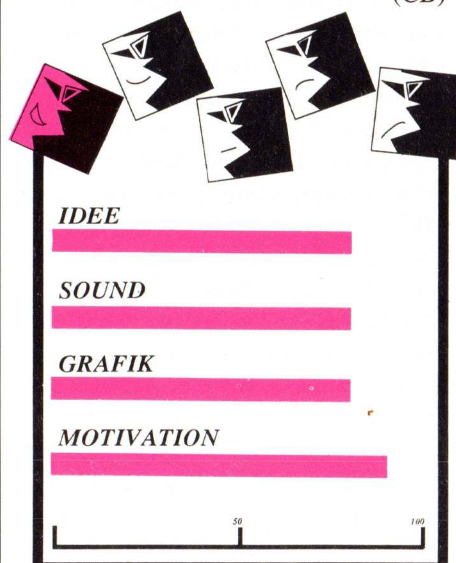
Noch beeindruckender als die Grafik ist der Sound. Soundprogrammierer Thomas Lopatic verdient ein großes Lob. Was er an Stereosound aus dem Amiga gekitzelt hat, ist hörenswert. Es gibt nicht nur eine fetzige Titelmelodie zu hören, die Soundeffekte während des Spiels sind ebenfalls gelungen. So ist während des Besuchs im Tempel die Flöte eines Schlangenbeschwörerers zu vernehmen. Die sechs Actionszenen sorgen für genügend Abwechslung. Spaß hat der Spieler angesichts der brillianten technischen Umsetzung. Bis auf eine Kleinigkeit gibt es bei "In 80 Tagen um die Welt" nichts zu bemängeln: Immer, wenn Fogg im Tempel von einem Bild in ein anderes läuft, ruckelt der Bildschirm, und die neue Grafik wird ins Bild gefahren. Wem dieser kleine Schönheitsfehler nichts ausmacht, der hat viele aufregende Stunden vor dem Monitor vor sich.

(CB)



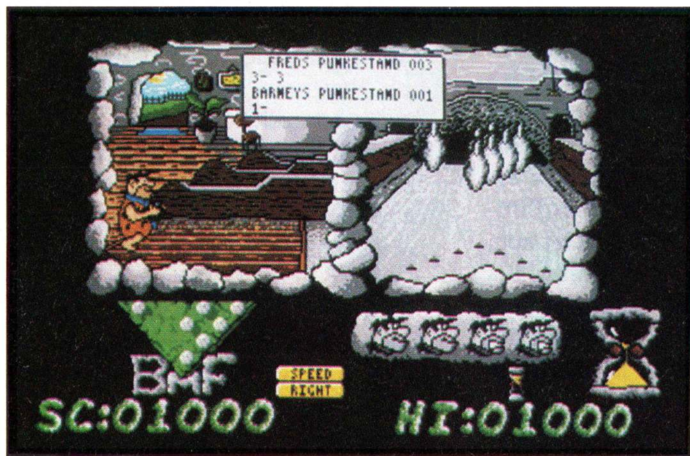
Das erste Abenteuer gilt es in Indien zu bestehen.

Eine menschliche Pyramide muß der Held in Japan bauen.



DIE FEUER-STEINS

Steinzeitcomic als Computerspiel



Der gute Fred schwingt die Bowling-Kugel.

Öffnet man die Verpackung des neuesten Spiels von Grand Slam Entertainment, FRED FEUERSTEIN, so findet man unter anderem ein Poster und einen Ansteckbutton (dieser wurde sofort von unserer neuen Mitarbeiterin entführt). Auf beiden Extras sind Fred Feuerstein und seine liebsten Freunde und Verwandten abgebildet. Fred's Frau Wilma, Töchterchen Pebbles, Freund Barney und dessen Frau Betty sind dem eifrigen Comicleser und Fernsehzuschauer alte Bekannte. Die beiden Familien müssen im Comic ihre Abenteuer in der Steinzeit bestehen.

Fred und sein Freund Barney möchten zum Bowling gehen. Doch Fred's Herzallerliebste hat andere Pläne mit Fred. Er soll gefälligst erst einmal die Wände streichen, bevor er seinem Vergnügen nachgeht. Fred fügt sich widerwillig den Wünschen seiner Frau.

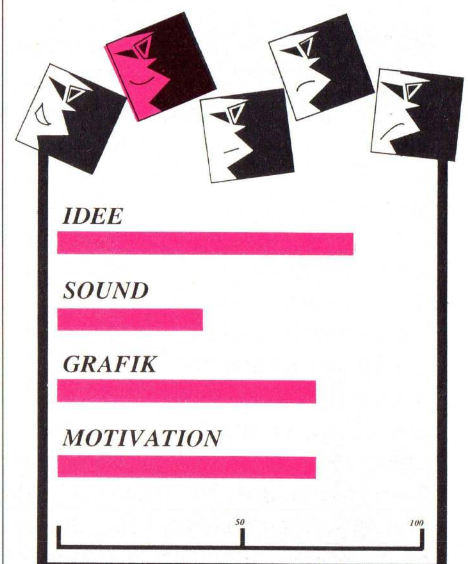
Als Spieler von "Fred Feuerstein" sollen Sie Fred unter die Arme greifen. Der Arbeitsvorgang ist stupide: Pinsel in den Farbeimer tauchen, zur Wand latschen, auf die Leiter steigen und per Feuerknopfdruck pinseln, bis der Pinsel keine Farbe mehr trägt. Töchterchen Pebbles bringt jedoch Leben ins Spiel. Manchmal springt sie aus ihrem Laufgitter und kritzelt Mondgesichter, Blumen oder ihren Namen an die frisch gestrichene Wand. Das Geschehen spielt sich unter großem Zeitdruck ab. Wenn fünf Sanduhren abgelaufen sind, kommt Wilma zu einer Inspektion. Falls die Wand nicht vollständig angemalt wurde, darf Fred noch einmal von vorne anfangen. Nach drei an der Zeit gescheiterten Versuchen bestraft Wilma ihren Gatten mit Babysitten, so daß der Arme nicht zu seinem Bowlingvergnügen kommt.

Im zweiten Level tragen Fred und Barney ein Radrennen auf dem Weg zur Bowling-Bahn aus. Beim Bowling im dritten Level treten Barney und Fred erneut gegeneinander an. Der Spieler muß seine Position auf der Bowling-Bahn und den Kraftaufwand einstellen, bevor er die Bowlingkugel loswirft. Nach den sportlichen Bemühungen muß man auf dem Nachhauseweg Pebbles suchen. Die Kleine ist aus ihrem Laufgitter entwichen und in Richtung Großbaustelle entfleucht. Fred muß sie dort in einem Konfusium an Gerüsten, Plattformen und Leitern finden, bevor sie sich verletzt.

"Fred Feuerstein" präsentiert sich in einer farbenfrohen, detaillierten Zeichentrickgrafik. Wenn man nicht wüßte, daß man es mit einem Computerspiel zu tun hat, könnte man GRAND SLAMs Spiel auch für einen Zeichentrickfilm halten. Die Grafiken sind dem Original sehr gut nachempfunden. Soundprogrammierer Benn Dagglish, vielen vielleicht noch aus ihren alten C64-Zeiten bekannt, hat die Titelmelodie aus der Fernsehserie zwar gut getroffen, jedoch hätte man mit den Fähigkeiten des Amiga den Sound noch interessanter gestalten können. Nach kurzer Zeit dreht man den Soundregler am Monitor entnervt auf "Leise".

Jede der vier Spielsequenzen besteht aus einem unterhaltsamen Gameplay, das den Spieler stundenlang an den Monitor fesseln kann. Der Schwierigkeitsgrad ist angenehm schwer. Kurzum, "Fred Feuerstein" ist durchweg ein gut gelungenes Computerspiel. Der Preis von 65 Mark ist durchaus gerechtfertigt, schließlich bekommt man auch noch zwei Extras in der Programmverpackung mitgeliefert. Was möchte das Spielerherz mehr? (CB)

Anbieter: IM



MEVILO

Die Adventure-Perle



In einem Gasthaus kann eingekehrt werden.

In MEVILO, dem neuesten Adventure vom französischen Softwarehaus COKTEL VISION, spielt man die Rolle eines berühmten Parapsychologen in einer aufregenden Geschichte. Als Spezialist in Sachen Gespenster und Zombies bekommen Sie einen Fall von Ihrem Klienten Michel Hubert Destouches.

Im Hause von Monsieur Destouches spukt es. Dumpfe Klopfgeräusche und unheimliche Seufzer versetzen den Armen in Angst und Schrecken. Ihr Klient ist der festen Überzeugung, daß es sich beim Inititiator der unheimlichen Vorgänge um einen Zombie handelt. Also reisen Sie an den Ort des Grauens, auf die Insel Martinique, um dem Geheimnis auf die Spur zu kommen. Dabei soll geklärt werden, woher der Zombie kommt und was er will.

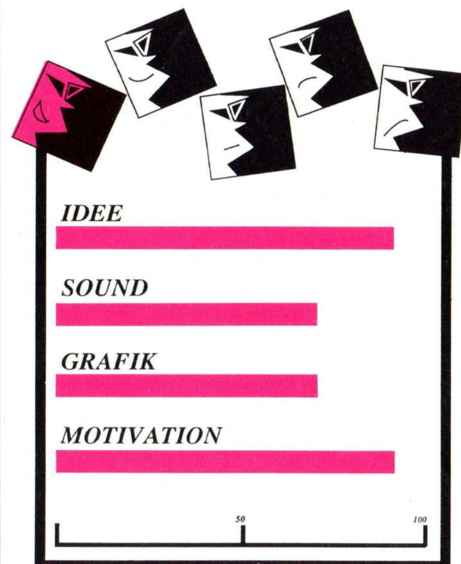
Während des Spiels sind Sie damit beschäftigt, Gegenstände zu untersuchen, Personen auszufragen und die Insel Martinique zu erkunden. Da kommt Adventurefieber auf. Keine Angst, bei Mevilo brauchen Sie nicht stundenlang auf der Tastatur herumzuhämmern, ein Großteil der Aktionen wird per Maus gesteuert. Wenn man einen Gegenstand untersuchen möchte, bewegt man den Mauszeiger auf ihn, und der Computer gibt in einem Fenster mit deutschem Text Auskunft, was es mit dem Gegenstand auf sich hat. Um den Ort zu wechseln, braucht man nicht "N", "S", "W" oder "E" einzugeben. Auf einer kleinen Karte am linken Bildschirmrand kann man mit Hilfe des Mauszeigers den gewünschten Ort aussuchen.

Auf den Spieler wartet eine Vielzahl von Überraschungen. Es gibt viele Rätsel zu lösen. So befindet sich auf der Insel beispielsweise ein alter Mann, der sich perfekt in der Geschichte des Landes auskennt. Der Insulaner stellt Ihnen Fragen über die Insel, die Sie im Multiple-Choice Verfahren richtig beantworten müssen. Die Fragen sind eine harte Nuß. Wußten Sie beispielsweise, daß man Begasse zur Produktion von Zucker verwendet? Durch genaues Lesen der Anleitung kommt man aber den richtigen Antworten auf die Spur. Wenn man dem alten Insulaner alle Fragen richtig beantwortet hat, schenkt einem dieser eine Flasche. Was es damit auf sich hat, soll nicht verraten werden.

Nach dem Rätselstündchen beim alten Insulaner schaute ich mir eine Plantage an und besuchte meinen Auftraggeber Monsieur Destouches. Dort sollte ich mit meinen parapsychologischen Fähigkeiten zu einer verstorbenen Frau Kontakt aufnehmen.

Mevilo präsentiert sich mit einer farbenfrohen Grafik. Soundeffekte, wie das digitalisierte Grunzen eines Schweins oder französische Musik sorgen für eine gelungene Atmosphäre. Coktel Visions neuestes Adventure versteht es, den Spieler für lange Zeit zu unterhalten. Mevilo ist eine Bereicherung für den Softwaremarkt. Durch das Spiel kann man viel über Martinique erfahren. Man lernt einen Teil der französchischen Kultur kennen, muß Indizien kombinieren und hat mit der spannenden Handlung viel Spaß. Kein Wunder, daß Mevilo schon eine hohe Auszeichnung bekommen hat. Der französische Premierminister Chirac verlieh der Autorin von Mevilo zur Veröffentlichung die silberne Medaille der Stadt Paris. Als Bonus liegt der Verpackung eine Cassette mit französischer Musik und ein Rezept für ein französchisches Spezialgericht bei. Liebhaber von anspruchsvollen Computerspielen kommen mit Mevilo voll auf ihre Kosten. (CB)

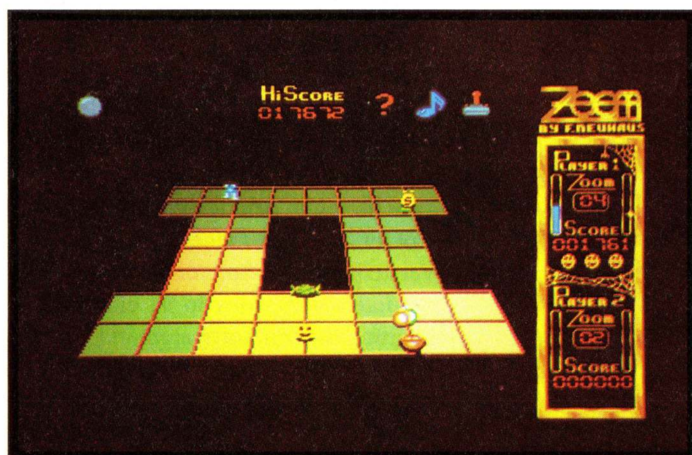
Info: Bomico



TOP 12 TIP

ZOOM

Schnell gerannt ist halb gewonnen



Überblick und schnelle Reaktion sind bei ZOOM gefragt, denn viele undefinierbare Wesen warten nur auf eine Verschnaufpause

Schon der Vorspann sorgt für gute Laune und stimmt auf das nachfolgende Spiel ein. Wie bei einer Theatervorführung erscheinen lustige Figuren, die nur aus einem runden Kopf, großen Händen und Füßen bestehen. Vor der stimmungsvollen Atmosphäre eines großen Publikums führen sie ihre Zauberkunststücke vor und treiben ihre Späße. Grafisch und akustisch ist dies ein hervorrageder Vorspann, in dem der Autor Frank Neuhaus und sein Grafikteam ihr Faible für Details zeigen.

Wenn man sich am Vorspann sattgesehen und -gehört hat, kann man auch spielen. Aus den unendlichen Weiten des Weltraums kommt das Spielfeld hervorrotiert. Das Spielprinzip ist einfach, das Spiel nicht: Auf einem schachbrettartigen Untergrund müssen die einzelnen Felder umrundet werden. Diese färben sich farbig, und man erhält 50 Punkte. 45 solcher Spielbretter stehen dem lustigen Spielbällchen bevor, dazu noch eine Anzahl Gegenspieler, die ihm das Dasein schwer machen. Besonders anhänglich sind die feuerroten Lippen, die sich dem Spieler ständig an die Fersen heften. Mehrere andere Wesen laufen ziellos in der Gegend herum und verwischen gemeinerweise die Umrandungen der Felder. Deshalb sollte man möglichst nur ausgefüllte Felder hinterlassen und alle herumlaufenden Wesen meiden.

Gesteigerte Aufmerksamkeit sollte man außerdem den kurzzeitig auftauchenden Gegenständen wie z.B. Würfeln, Kraftpillen u.ä. schenken, die außer zu Bonuspunkten auch zur Lähmung der Gegner führen können. In höheren Leveln kommen schwierigere Spielfelder, unzählige Gegenspieler und neue Hindernisse wie z.B. plötzlich auftauchende Löcher und rotierende Objekte hinzu. Teilweise sind die Wege des Spielfelds nur noch ein Karo breit und so viele Objekte unterwegs, daß man kaum eine Chance zum Weiterkommen hat. Da gilt es, Nerven zu bewahren und ja nicht kopflos herumzurennen, sondern auf eine Lücke im Engpass zu warten.

ZOOM ist ein sehr spannendes und unterhaltsames Spiel. Wer einmal damit angefangen hat, die lustige Kugel über die Karo-Ebenen zu steuern, der wird davon nicht mehr so schnell ablassen wollen. Durch die ansprechende Grafik und die originalen Soundeffekte wird die hohe Motivation von ZOOM noch unterstützt, so daß man dieses gelunge Spiel sehr empfehlen kann. (MN)

Bezugsquelle:
Discovery Software, Würzburg



INSERENTEN-VERZEICHNIS

Einkaufsführer

Hier finden Sie Ihren Commodore/Amiga Fachhändler

1000 Berlin



RUNOW
Büroelektronik
Keithstraße 26 · 1000 Berlin 30
☎ 26 111 26

COMPUTER-STUDIO
Schlichting
. . . die etwas andere Computerei
ATARI-Fachmarkt
MS-DOS Fachmarkt · NEC-Fachhandel
Katzbachstraße 8 · 1000 Berlin 61
☎ 030/7864340

2000 Hamburg

Bit Computer Shop
Osterstraße 173 · 2000 Hamburg 20
Telefon: 040/494400

Createam
Computer Hard & Software
Bramfelder Chaussee 300 · 2000 Hamburg 71
Telefon: 040/6416473+6416861

Gerhard u. Bernd Waller GbR
Computer und Zubehör-Shop
Kieler Straße 623
2000 Hamburg 54
Telefon (040) 5706007 + 5705275

2000 Hamburg

GMA mbH
040/2517416/17
Systemhändler
Wandsbeker Chaussee 58
2000 Hamburg 76

Computer
Hardware · Software · Zubehör
Lilienstraße 32
(beim Mönckebergbrunnen)
2000 Hamburg 1
Tel. (040) 336708



SYSTEMSHOP ®

RADIX Bürotechnik
Heinrich-Barth-Straße 13
2000 Hamburg 13
Telefon (040) 441695
NEU: Software Shop

2160 Stade

BERGAU
Büromaschinen · EDV-Systeme
Neue Straße 5 · 2160 Stade
Telefon (04141) 2364 u. 2384

2390 Flensburg

ecl electronic computer laden ohg
Norderstraße 94-96 · D-2390 Flensburg
☎ (0461) 28181 + 28193

2900 Oldenburg

GOLDT
Computerhaus
Donnerschweer Straße 127-129
(gegenüber Weser Ems Halle)
2900 Oldenburg
Telefon (0441) 884706

2940 Wilhelmshaven

Radio Tiemann
Commodore-Systemfachhändler
Marktstraße 52
2940 Wilhelmshaven
Telefon (04421) 26145

3000 Hannover

COM DATA
Am Schiffgraben 19 · 3000 Hannover 1
Telefon (0511) 326736

3000 Hannover

CSJ
COMPUTERSOFT GmbH
HANNOVER'S SOFTWARETHEK NR. 1
IBM PC AMIGA ATARI
An der Tiefenriede 27 · 3000 Hannover 1
Tel. (05 11) 88 63 83 · 24 Stunden Service

3500 Kassel

Hermann Fischer GmbH
Commodore-Systemfachhändler

Rudolf-Schwander-Str. 5-13
3500 Kassel
Telefon (05 61) 70 00 00

Erfolgreich werben im Amiga-Einkaufsführer
Sprechen Sie mit uns.
Heim-Verlag ☎ (0 61 51) 5 60 57 BUF

4650 Gelsenkirchen-Horst



Hard- und Software, Literatur
Bauteile, Service, Versand
Groß- und Einzelhandel
Poststraße 15 · 4650 Gelsenkirchen-Horst
Telefon (02 09) 5 25 72

5000 Köln

BÜRO MASCHINEN **braun**
AM RUDOLFPLATZ GmbH
5000 KÖLN 1
RICHARD-WAGNER-STR. 39
TEL. (02 21) 21 91 71

6000 Frankfurt

COMPUTER **bco**
Büro-Computer + Organisations GmbH
Oederweg 7-9
6000 Frankfurt/Main 1
☎ (0 69) 55 04 56 - 57
Commodore OKI ATARI TOSHIBA

6200 Wiesbaden

DTM
Poststraße 25
6200 Wiesbaden-Bierstadt
Telefon (0 61 21) 56 00 84
Telefax (0 61 21) 56 36 43
Werbung und EDV GmbH
AUTORISIERTER COMMODORE SYSTEM-HÄNDLER
Commodore

6380 Bad-Homburg

PDC GmbH
Produkte u. Details Computerverband
Louisenstraße 115
Ladenpassage Alter Bahnhof
6380 Bad-Homburg
Telefon (0 61 72) 2 47 48
Autorisierter Commodore-Systemhändler!

6457 Maintal

Landolt-Computer
Beratung · Service · Verkauf · Leasing
Autorisierter Commodore-Händler
Wingertstr. 112 · 6457 Maintal/Dörnigheim
Telefon (0 61 81) 4 52 93

6551 Fürfeld

MW **MICHAEL WEISGERBER**
HARD + SOFT
Rathausstraße 2
6551 Fürfeld
Telefon (0 67 09) 7 78

6680 Neunkirchen

SHOP 64
Computer GmbH
Saarbrücken
Saarlouis
Homburg
St. Ingbert
Neunkirchen ☎ 0 68 21 / 2 37 13
Commodore Systemhändler

6700 Ludwigshafen

COMPUTING & SOUD
Dieter Hieske
Schillerstraße 36 · 6700 Ludwigshafen
Telefon (06 21) 67 31 05
C — H — S
AMIGA USER ONLY

6800 Mannheim

GAUCH + STURM
Computersysteme + Textsysteme
6800 Mannheim 24
Casterfeldstraße 74-76
Telefon (06 21) 85 00 40 · Teletex 6 211 912

7000 Stuttgart

»If AMIGA, go to Schreiber«
Stuttgart's starker Computer-Laden.
SCHREIBER COMPUTER
Rotebühlplatz 10
7000 Stuttgart-1
Tel. 0711/227099
Neu · Neu · Neu · Neu
Im SUBWAY
BREUNINGER City

7140 Ludwigsburg

B D T
BÜRO-DATEN-TECHNIK-VERTRIEBS GMBH
Kurfürstenstraße 18 · 7140 Ludwigsburg
Telefon (0 71 41) 2 50 74

7890 Waldshut-Tiengen

hettler-data
service gmbh
Lenzburger Straße 4
7890 Waldshut-Tiengen
Telefon (0 77 51) 30 94

7900 Ulm

EDV-Systeme
Software-erstellung
Schulung
Systemhaus:
Frauenstraße 28
7900 Ulm/Donau
Tel. (07 31) 2 80 76
Telex 7 12 973 csulm-d
COMPUTER STUDIO GmbH

8000 München

ludwig
COMPUTER + BÜROTECHNIK
COMPUTER · SOFTWARE · PERIPHERIE
BERATUNG · TECHN. KUNDENDIENST
INGOLSTÄDTER STRASSE 62L
EURO-INDUSTRIE-PARK · 8000 MÜNCHEN 45
TELEFON 089/3 11 30 66 · TELETEX 89 83 41

AMIGA


MACHINE
Computergraphic
Gabriele Lechner
Beratung
Schulung
Verkauf
Planeggerstraße 1 · 8000 München 60
Telefon (0 89) 8 34 05 91

8400 Regensburg

Zimmermann elektroland
8400 Regensburg
Dr.-Gessler-Str. 8
☎ 09 41 / 9 50 85
8390 Passau
Kohlbruck 2a
☎ 08 51 / 5 20 07

Bei uns werben bringt
GEWINN
Sprechen Sie mit uns.
Heim Verlag ☎ 0 61 51 / 5 60 57 BUF

8700 Würzburg

SCHOLL BÜROTEAM

Hardware · Software
Service · Schulung

computer center

am Dominikanerplatz
Ruf (09 31) 3 08 08 - 0

Schweiz

CH-8021 Zürich



Langstrasse 31
Postfach
CH-8021 Zürich

Tel. (01) 2 41 73 73
Telex 8 14 193 seco

CH-4054 Basel

Wir sind Amiga-Freaks.

Unsere Öffnungszeiten:
Von Dienstag bis Freitag
9.30—12.30 und 14.00—18.30 und
am Samstag 9.30—16.00 Uhr

SYSAG COMPUTERCENTER

Basel: Tel. 061/39 25 25 · Holeestrasse 87 · 4054 Basel
Aarau: Tel. 064/22 63 33 · Kasernenstrasse 26 · 5000 Aarau



Die Firma, die Euch MarauderII® und GRABBIT® gebracht hat, hat jetzt in **Deutschland** ein Büro. Schaut mal nach unseren neuesten Titeln.

Friedrich-Spee-Straße 11
8700 Würzburg
Telefon (09 31) 88 48 22

PROBLEME MIT ENGLISCHER ANLEITUNG?

Übersetzen von englischen Bedienungsanleitungen, Computerhandbüchern und Programmen für AMIGA, Atari, Macintosh, MS-DOS, Unix-Xenix, Periferiegeräte und andere durch *Spezialisten.*

Sonderangebote deutscher Anleitungen für AMIGA

Championship Golf	**30,–**	**DBW Render**	**20,–**
Flight Simulator II	**35,–**	**Digal doc Kurzanleitung**	**15,–**
Starglider	**20,–**	**ASDG-RRD Ram Disk**	**15,–**
Digi-View V 2.0 NTSC/PAL	**30,–**	**NEWZAP V 3.0 Diskettenmonitor**	**15,–**
Sound Scape Sampler	**20,–**	**PrtDrvGen Druckertreibergenerator**	**15,–**
The Music Studio	**40,–**	**Expert Cardridge**	**15,–**

Bei Versand im Inland berechnen wir DM 5,– für Porto Verpackung/NN.
Versand ins Ausland nur mit Vorauskasse + DM 4,– für Porto/Verpackung



0 83 74 - 98 73

T. Sonnenmoser · Hauptstraße 26 · D-8961 Haldenwang

AMIGA ★ Public-Domain ★ AMIGA

STEFAN OSSOWSKI

Ca. **550** Disketten lieferbar: Fisch 1—**138**, Panorama 1—**68**, Fang 1—**51**, Amicus 1—**20**, Auge 4000 1—**18**, Taifun 1—**60**, Chiron Conception 1—**68**, RPD 1—**113**, KICK 1—**60** u.v.a.

Bei Bestellung von mindestens **10** Disketten wird die PD-Disk **CLI-Help** – unentbehrlich für Anfänger und Einsteiger – **Kostenlos** mitgeliefert! – Stichwort: **CLI-Help**

Einzeldisk DM 7,00
ab 10 Stück DM 6,50
ab 20 Stück DM 6,00
ab 30 Stück DM 5,50
ab 50 Stück DM 5,00
ab 100 Stück DM 4,70
ab 200 Stück DM 4,50

Wir kopieren selbstverständlich auf **2-DD**-Disketten!

★★ Taifun ★★
Super-PD-Software
Taifun = Auslese der besten auf dem Markt befindlichen PD-Programme.
Eigenentwicklung – Exclusivvertrieb
Sonderangebot:
Nr. 1–Nr. 30 V-Scheck DM 160,–
Nr. 1–Nr. 40 V-Scheck DM 205,–
Nr. 1–Nr. 50 V-Scheck DM 245,–
Die **echten Taifun** erkennt man an der Original-Seriennummer!

2 Katalogdisks mit Kurzbeschreibung aller Programme gegen **DM 5,–** (Scheck/Briefmarken) anfordern!
Garantie: Versand erfolgt am gleichen Tage des Bestelleinganges!

Das große Public-Domain-Buch
Ausführliche deutsche Beschreibung zu ca. 100 Public-Domain-Programmen auf ca. 320 Seiten.
DM 49,– zuzüglich Versandkosten

10 % Abo-Rabatt bei Neuerscheinungen. (Alle oder bestimmte Serien)

Super-Grafik-Paket
Inhalt: Ray-Tracing (DBW-Render), 1 Zeichenprogramm, 1 Spriteeditor, 28 Zeichensätze, Apfelmännchen, Fractals, viele IFF-Grafiken, Slideshow-Programme...
Sonderpreis: DM 62,– Scheck **DM 66,–** Nachnahme
– incl. Porto und Verpackungskosten –

STEFAN OSSOWSKI
IHR PD-SPEZIALIST

Veronikastraße 33
D-4300 Essen 1
Tel.: 02 01 / 78 87 78

2 Jahre Garantie auf Hardware und geprüfte Software mit Zertifikat, beides in bester Qualität unterscheiden uns zu Billigangeboten

AMIGA aktuell — Dieter Hieske
Ladenlokal Schillerstraße 36
6700 Ludwigshafen-Oggersheim
Telefon (06 21) 67 31 05

Öffnungszeiten: Montag—Freitag 9.30—12.00 Uhr, Montag—Freitag 12.00—18.00 Uhr, Samstag 9.00—13.00 Uhr. Danach Anrufbeantworter.

Public Domain Software Kopierservice
lieferbar sind: Fish 1-128, Panorama 1-51, Faug 1-53, Amicus 1-20, Amuse 1-3, Chiron 1-39, TBAG 1-7, Auge 4000 1-14, Kickl 1.2V 1-30, ACS 1-39, Safe 1-4, Tornado 1-30, Taifun 1-50 usw.

Kopierpreise, wenn wir die Disketten 2DD mitliefern — Preise per Stück: 1-9 Disk 7,00 / 10-19 6,50 / 20-29 6,00, 30-49 5,50 / 50-99 5,00 / 100-200 4,50 ab 200 4,00, Fishserie 1-128 DM 539,00 (4,21)

Kopierpreise, wenn Sie uns Ihre Disketten per Einschreiben liefern — Preise per Stück: 1-9 4,00 / 10-19 3,50 / 20-29 3,00 / 30-49 2,50 / 50-99 2,00 / 100-199 1,50 ab 200 1,00.

Software Anwender
Sculpt 3D 182,90 / Animate 3D 298,00 / Forms in Flight 158,95 / Pagesetter D 298,00 / Videotitler 178,00 / Butcher D 99,00 / Photon Paint 239,00.

Software Spiele
StripPoker II + DM 42,95 / Destroyer DM 68,90 / Giana Sisters DM 51,90 / Ferrari Formula 1 DM 76,95 / BMX Simulator DM 54,90 / Pink Panther DM 54,90 / Vampires Empire DM 54,90 / Doubble Bubble DM 54,90 / Strike Force Harrier DM 67,50 / Winter Olympic 88 DM 54,00. Neuheiten täglich, bitte anrufen. Es lohnt sich.

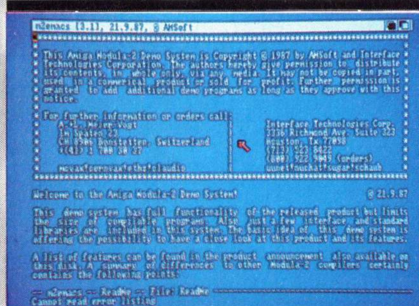
Hardware
A500: 512 Kb Speicher, Uhr absch. DM 249,00, A1000: 2 MB ext. Box DM 999,00, A500: 2 MB ext. Box DM 999,00, Laufwerk 3,5" DM 339,00, A500: Bootselektor DF1 und DF2 je 19,95

Versand per Nachnahme + DM 8,00 Porto, Versand per Vorauskasse ohne Versandkosten.
Preisliste und Public Domain Inhaltsverzeichnis (70 Seiten) kostenlos.

DIE NEUEN

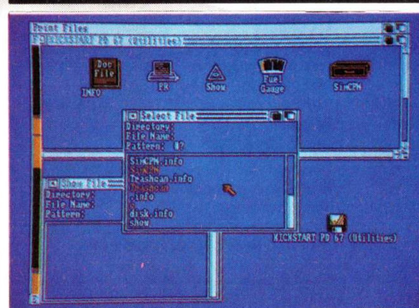
*Liebe Leser,
wieder einmal haben wir die neuesten und besten PD-Programme für die KICKSTART PD-Sammlung zusammengestellt. Langsam gibt es neben der Programmflut aus den englischsprachigen Gebieten aber auch hier einige Autoren, die gute PD-Programme schreiben. Bitte denken auch Sie daran, wie nützlich und interessant solche Programme für die AMIGA-Benutzer sein können. Lesen Sie deshalb auch unseren interessanten Aufruf. In der Hoffnung, daß auch Sie sich rege an dieser Aktion beteiligen, wünsche ich Ihnen bis zum nächsten Mal viel Erfolg beim Progammieren Ihrer Programme.*

PD 66: M2-Modula



M2 AMIGA: Dies ist die funktionsfähige Demo-Version des M2-Modula-Compilers. Beschränkungen gibt es nur hinsichtlich der Größe der compilierbaren Programme, außerdem sind nur wenige Schnittstellen und Standard-Bibliotheken vorhanden.
Auf der Diskette sind einige Demo-Programmme enthalten, die z.T. direkt für diesen Compiler entwickelt wurden.

PD 67: UTILITIES



"was mer hat ..." - sicherlich kennen Sie diesen Spruch mittlerweile und wie immer stellen wir auf diesen Disketten die besten Hilfsprogramme zusammen. Unter anderem sind dies:
PR: ein Programm zum Ausdrucken von Texten. Das Inhaltverzeichnis wird in einem Fenster dargestellt und das File kann mit der Maus ausgewählt werden. Auf dem Ausdruck erscheint dann eine fortlaufende Seitennummerierung, der Name und das Datum.
SHOW: entspricht dem Programm PR jedoch werden die Texte nur angezeigt.
FUEL GAUGE: ist eine grafische Anzeige für den vorhandenen und benutzen Speicherbereich.
SIMCPM: CP/M-Emulator, der den 8080-Prozessor emuliert. Mit H19 Terminal.

PD 68: GRAFIK-DEMOS

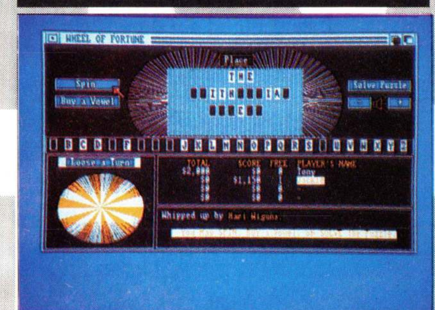


WBLANDER: versuchen Sie das Raumschiff sicher auf einem Workbench-Fenster zu landen. Drei Triebwerke können zum Abremsen und Festlegen der Richtung eingesetzt werden. Bei erfolgreicher Landdung wird der Kontrolltower eingeblendet, der Ihnen zur Landung gratuliert. Auch dieses Programm hat am BADGE KILLER DEMO CONTEST teilgenommen.
MACHINE: eine hervorragende Demo von Allen Hastings erstellt mit VidoeScape 3D.
DRUNKENMOUSE: wenn Sie nicht betrunken sind, dann liegt es sicherlich an diesem Programm, daß Sie Schwierigkeiten haben, etwas anzuklicken.

PD 69: GRAFIK-DEMOS

AMUC_DEMO: Ein Bild mit 200 x 2000 Bildpunkten scrollt vertikal über den Bildschirm. Ein sehenswertes Erlebnis.
HBHILL: Dies ist Kevin Sullivans Beitrag zum BADGE KILLER DEMO CONTEST. Die Demo nutzt den Extra-Half-Brite-Modus (64 Farben) des AMIGA, der jedoch auf sehr alten Modellen nicht vorgesehen war.
HAMmmm: Eine mit Musik unterlegte Linendemo von Phil Burk. Inclusive Source-Code in JForth.
STARS: Ein interessanter Flug durch einen Sternenhimmel.
WIREDEMO: zeigt einen rotierenden Linienkörper, mit Source in C.

PD 70: SPIELE



CLUE: das bekannte Brettspiel für pfiffige Detektive in einer ansprechenden Computerversion. Verschiedene Räume müssen untersucht, Personen befragt und die Ereignisse kombiniert werden, um den Mörder herauszufinden. Die gute Grafik und die einfache Mausbedienung machen eine Anleitung weitgehend unnötig.
WHEEL: Bei diesem Spiel muß ein Ausdruck erraten werden. Diese Glücksrad-Simulation ist grafisch gut aufgemacht und komplett mausgesteuert. Da die zu erratenden Begriffe in englisch sind, ist dieses Spiel gut zum Erlernen dieser Sprache geeignet.
MASTERMIND: bekanntes Kombinationsspiel, bei dem eine Farbkombination erraten werden muß. Mausgesteuert!
RISTINOLLA: Eine spielstarke GO-Muko-Variante. Mausgesteuert und sehr schnell!

WANTED !!!

Deutsche PD-Software gesucht!
Belohnung: 5 PD-Disketten
Wer aufmerksam das PD-Angebot studiert, wird schnell feststellen, daß fast alle Programme und auch die meisten Grafik- und Sounddemos von amerikanischen und englischen Programmierern erstellt werden. Dies hat unter anderem zur Folge, daß fast alle Kommentare in den Programmen und vor allem die Anleitungen in englisch verfaßt sind. Dies macht es manchem deutschen AMIGA-Einsteiger nicht leicht, den AMIGA und dessen Programmierung kennenzulernen. Wir wollen deshalb noch einmal alle Programmierer im deutschsprachigen Raum aufrufen, ihre Programme, die sonst in einer dunklen Schublade verschwinden, in die PD-Sammlung zu geben.
Als kleinen Anreiz erhält jeder Programmierer, dessen Programm in den KICKSTART PD SERVICE aufgenommen wird, fünf PD-Disketten seiner Wahl.

DIE KLASSIKER

Im folgenden finden Sie eine, nach Sachgebieten sortierte Auswahl des Angebots. Sollten Sie an einer kompletten Liste interessiert sein, dann senden Sie einen frankierten und an Sie adressierten Rückumschlag an die untenstehende Adresse.

UTILITIES:

QuickCopy (gutes Kopierprogramm), DirUtil (File-Copy), FileZap (File-Monitor), DiskZap (Disk-Monitor), Disk-Salv (Diskettenretter), System-Monitor, CSH (UNIX-ähnliche Shell) **[PD 7]**

CLOCK, PORTAR, MACView, Kickbench, Disassembler, Tracker, Checkmodem, POPCLI und vieles mehr. **[PD 25]**

Auf dieser Diskette befinden sich einige sehr schnelle und leistungsstarke Editoren (UEDIT, MED, BLITZ) mit dazugehörigen Zeichensatz-Utilities. **[PD 28]**

PrtDrvGen (erstellt Drucker-Treiber), DropShadow (jedes Fenster bekommt einen Schatten), MemClear (löscht den Speicher), ScreenSave (speichert den Bildschirm auf Diskette), Compress (komprimiert Programme) **[PD 29]**

ASDG (resetfeste RAM-Disk), FixDisk, ErrorCk (zur Fehlersuche auf der Diskette), DiskCat (erstellt eine Übersicht über die Programme Ihrer Disketten) **[PD 35]**

AddMem (zum Konfigurieren von Speichererweiterungen), Mem-View (zeigt den Speicherinhalt als Grafik an), GetRom, MegaPatch (paßt die Kickstart des AMIGA 1000 für das autom. Erkennen von internen Speichererweiterungen an) **[PD 37]**

Alles, was Sie zu dem von ELECTRONIC ARTS entwickelten Grafik-Standard (IFF-Format) wissen müssen: Laden, Speichern, Komprimieren, Dekomprimieren. Mit Dokumentationen und Source-Codes in C. **[PD 41]**

SECTORAMA (ein sehr nützlicher Disketten- und Festplatten-Monitor, mit dem verlorene oder zerstörte Daten wiederhergestellt werden können), SILICON (ein sehr komfortabler CLI-Ersatz mit separatem Ausgabe- und Eingabefenster), DBUG (maschinenunabhängiger Debugger von Fred Fish (Source in C)), TIMER (eine Stoppuhr für die Workbench) **[PD 47]**

CONMAN V1.0 (sehr nützlicher CLI-Ersatz), MOUSEREADER (Texte lesen mit Mausbedienung), TIMERAM, MEMWATCH, DISKMAN V1.0 (Bedienung fast aller Diskettenoperationen mit der Maus, sehr umfangreich!), DIRMASTER V1a (sehr schönes Disketten-Verwaltungsprogramm, viele Funktionen) **[PD 52]**

Einige schöne Grafikdemos und Utilities zu diesem Thema **[PD 55]**

GOMF (bekämpft den GURU), VCHECK (Viruschecker von Commodore), Journal (Mausaktionen werden aufgenommen), PRINTER-STEALER (leitet Druckerausgabe auf Disk um), Editordemo **[PD 57]**

Mit diesen ICON-TOOLS ist es jedem möglich, eigene Icons zu erstellen und zu verändern. Diese können sogar gegen alte ausgetauscht werden: DPI-CON ermöglicht es, eine mit Deluxe Paint erstellte Brush beliebiger Größe in ein Icon umzuwandeln. SNAP wandelt einen beliebigen Bildschirm in ein Icon. AL-TICON fügt zwei Icon-Bilder zu einem 'animierten' Icon zusammen. Und viel schöne Icons.**[PD 65]**

Sprachen:

XLISP 1.7 (neueste Version) mit ausführlicher Anleitung (über 50k) **[PD 16]**

MODULA-2: Pre-Release eines Modula-Compilers mit verschiedenen kleineren Beispielprogrammen, die als Source-Code vorliegen **[PD 17]**

MVP-FORTH und C-FORTH (C-Forth ist ein recht leistungsfähiger FORTH-Interpreter, der auch als Quelltext vorliegt) **[PD 22]**

Eine Diskette voll Programmen (Spiele, Grafikprogramme uva.) in AMIGABASIC zum Reinschauen, Verändern, Lernen. **[PD 50]**

Ein einfacher C-COMPILER, in den man etwas Arbeit stecken muß, damit er läuft. Eignet sich für Interessenten am Compilerbau, da der Source-Code vorliegt. **[PD 51]**

Eine Sammlung von Programmen, die besonders dem Anfänger zeigen, wie man Intuition programmiert. Die Programme liegen sowohl aus C-Quellcode als auch als fertige Programme vor, die sofort gestartet werden können. **[PD 1]**

ASM68K (Macro Assembler mit guter Dokumentation), ASM (68010 Macro Assembler wie im AmigaDOS Manual beschrieben), BLINK (bekannter Linker), AS6502 (portabler 6502-Assembler mit Source in C)**[PD 56]**

ANWENDUNGEN:

MCAD ist ein wirklich gut gemachtes CAD-Programm, daß jedoch nur im Interlace-Modus läuft. Es bietet die einfachen Zeichenfunktionen und Features wie Zoom, Group, Ungroup, Grid, Move, Rotate). Auf der Diskette sind mehrere Dokumente, die das Programm erklären. **[PD 36]**

MICROSPELL (überprüft die Rechtschreibung), ACCESS (Terminal), QBASE (Dateiverwaltung) und einige andere Anwenderprogramme **[PD 54]**

GRAFIK & ANIMATION:

RAY TRACERS: wunderschöne räumliche Bilder, die auf einer VAX berechnet und auf den AMIGA übertragen wurden **[PD 11]**

Auf zwei randvollen Disketten erleben Sie eine einmalige Dia-Show mit hervorragend digitalisierten futuristischen Bildern in voller PAL-Auflösung. Dazu gibt es stimmungsvolle, sphärische Musik. (IFF-Format)**[PD 26 & PD 27]**

Boing!, Rotate, Sparks, Moire, Dazzle, 3DCube, Scales, Sizzlers. Sehenswert ist der Film 'Atari meets AMIGA', der die erste und einzige Begegnung der beiden Computer dokumentiert. Sehr schön ist das Programm LANDSCAPE, das wunderschöne fraktale Berg- und Tallandschaften erzeugt.**[PD 40]**

Eine einmalige Show, bei der eine digitalisierte Katze in gleitenden Bewegungen über den Bildschirm trabt. Erstellt wurde diese faszinierende Animation mit einem Digitizer, DPaint und VideoScape 3D.**[PD 43]**

Eine weitere Diskette (siehe auch PD 42) mit phantastischen RAY-TRACER-Bildern, unterlegt mit digitalisierter, fetziger Musik. (IFF-Format)**[PD 46]**

Die neuesten Demos vom BADGE DEMO KILLER CONTEST: hervorragende Grafik und Animation. **[PD 58]**

Mit drei herrlichen Grafik-Animationen stellt Eric Graham seine Programme Sculpt 3-D und Animate 3-D vor. Die Bilder der Animationen sind nach dem Ray-Tracer-Verfahren berechnet.**[PD 59]**

RAY-TRACER DBW-Render ist ein leitungsfähiges Programm zum Erstellen von Bildern nach dem Ray-Tracing-Verfahren. Die Berechnung der Bilder dauert mehrere Stunden, aber die Ergebnisse sind hervorragend. **[PD 60]**

Auf dieser Diskette sind die Beiträge von Allen Hastings, Hobie Oris und Phil Burk zum BADGE KILLER DEMO CONTEST enthalten. RIPPLES, WAVE-BENCH, VIACOM und DROPSHADOW 2.0. **[PD 61]**

HAGENDEMOS: Die Filme, die Joel Hagen zeigt, sind unbestritten Kunstwerke, denn nicht umsonst wurde "RGB" der Gewinner des BADGE KILLER DEMO CONTEST. (benötigt 1 MB RAM !)

BULLY ist der absolute Wahnsinn, denn was sich hier auf drei Workbench-Screens abspielt ist wirklich einmalig. LI-NEDRAWER, POPLIFE **[PD 62]**

Diverses:

AMIGA-Tutor, eine Einführung in die Bedienung des AMIGA 500. Ein farbenfroher Lehrgang, der ganz am Anfang beginnt und mit vielen Bildern und Grafiken die Grundbegriffe des AMIGA erklärt. (für Anfänger, komplett in deutsch)**[PD 21]**

SPIELE:

YachtC (Würfelspiel für 4 Personen), Puzzle, Missile (verteidigen Sie Ihre Stadt, starker Sound), TriClops (sehr schönes 3D-Spiel), Breakout (3D-Effekt mit Brille), Trek73 (bekannte Star Trek-Variante) **[PD 2]**

TUNNEL VISION: Werden Sie den Weg durch das Labyrinth finden? REVERSI: eine spielstarke Version des bekannten Brettspiels KLONDIKE: ein Patiencе-Kartenspiel **[PD 34]**

Ein echter Leckerbissen für Spielefans ist das Grafikadventure LARN. In unüberschaubaren unterirdischen Gängen müssen Gold und Schätze gesucht werden, die von Gnomen und Vampiren bewacht werden. Wirklich sehr empfehlenswert! **[PD 44]**

GRAVITYWARS (interessantes Weltraumspiel, bei dem sich zwei Raumschiffe im Kampf gegenüberstehen, viele Optionen!), OTHELLO (sehr schöne Reversi-Variante mit eigenem Fenster!), STREITPATIENCE (Patience-Variante von Hellmut Voelcker), CHESS (spielstarkes Schachprogramm)**[PD 45]**

AMOEBA (sehr gute SPACE INVADERS-Adaption), ASTEROIDS (ein Weltraumspiel, das ebenfalls von Spielhallencomputern umgesetzt wurde), EGYPTIAN RUN (ein Geländewagen muß durch die Wüste gesteuert werden)**[PD 63]**

BACKGAMMON (das bekannte Strategiespiel), PUSH OVER (versuchen Sie durch Schieben fünf Steine in ein Reihe zu bekommen), PUZZLE PRO, IFF2PICS (zwei Programme, die ein beliebiges IFF-Bild in ein Puzzle verwandeln, mausgesteuert!) **[PD 64]**

Die Programme laufen auf allen AMIGA-Computern mit Kickstart/Workbench 1.2, allerdings sollten mindestens 512k Speicher vorhanden sein (einige Grafik-Demos brauchen 1MB!). Sollten dennoch Einschränkungen gelten, wird dies bei den betreffenden Programmen angegeben.

Versandbedingungen:

Um einen schnellen und problemlosen Versand zu gewährleisten, beachten Sie bitte folgende Punkte:

- Für jede Diskette ergibt sich ein Unkostenbeitrag von DM 10.
- Pro Sendung kommt ein Versandkostenbeitrag (für Porto und Verpackung) von DM 5.- (Ausland DM 10.-) hinzu.
- Bei Nachnahme zuzüglich DM 3.70
- Bitte legen Sie, falls zur Hand, einen Aufkleber mit Ihrer Adresse bei

Anschrift:

MERLIN Computer GmbH
KICKSTART Redaktion
Postfach 55 69
6236 Eschborn

Die Diskettenbestellung kann auch telefonisch erfolgen. Der Versand erfolgt dann per Nachnahme.
Tel.: 06196/ 48 18 11 (MO.-FR. von 9-17 Uhr)

VORSCHAU

Hardware

Soundsampling ist ein Thema, das gerade für AMIGA-Besitzer ein Reizwort ist. Wir haben einige neue Geräte im Test und möchten Ihnen unsere Erfahrungen weitergeben.

MS-DOS

Batch-Dateien unter MS-DOS. Mehr als man denkt.

Kickstart 1.3 was nun? Fragen sich viele Besitzer eines AMIGA 2000 oder 500. Einige Firmen bieten Umschaltplatinen an mit denen sich dieses Problem beheben läßt. Wie funktioniert's ?

Software

Festplatten Backup für AMIGA-Platten. Wir stellen Ihnen zwei Kandidaten vor. Wie sicher sind Ihre Daten bei einem Backup? Wie komfortabel arbeiten solche Programme auf dem AMIGA?

Grundlagen

Die Zeichenkiste. Neue Perspektiven mit dem AMIGA.

Impressum

KICKSTART

Chefredakteur:

Uwe Bärtels (Chefredakteur)(UB)
Markus Nerding (Stellvertreter)(MN)

Redaktion:

Andreas Krämer (AK)
Gerald W. Carda (GC)
Harald Schneider (HS)
Marcelo Merino (MM)
Harald Egel (HE)

Herausgeber:

'MERLIN'-Computer GmbH
Industriestraße 26
Postfach 5569
6236 Eschborn
Tel.: 06196/481811
FAX: 06196/41137

Redaktionelle Mitarbeiter:

Christian Keller (CHK)
Andreas Diebold (AD)
Wolf Dietrich (WD)
Andreas Suchy (AS)
Jobst Hermeier (JH)
Carsten Borgmeier (CB)

Autoren dieser Ausgabe:

Sebastian Dosch
Roland Foerster
Garry Glendown
Jobst Hermeier
Raymund Hofmann
Herbert Kunz
Cristoph Münzer
B. von Rüschen
Frank Schäfer
Jürgen Schmidt
Lars Schmitt

Public Relations:

Claus Peter Lippert

Auslandskorrespondenz:

D. dela Fuente (GB)
L. Hennely (USA)

Verlag:

Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt 13
Tel.:06151/56057
FAX:06151/55689
FAX:06151/56059

Verlagsleitung:

Hans-Jörg Heim

Anzeigenverkauf:

Kyriakulla Margaritis
Uwe Heim (Ltg.)

Anzeigenpreise:

nach Preisliste Nr.3, gültig ab 1.1.88

Grafische Gestaltung, Titelbild, Satz:

Fabian & Mayer

Produktion:

Karl-Heinz Hoffmann

Fotographie:

Rainer Spirandelli, Archiv

Titelfoto:

Fabian & Mayer

Kostümverleih:

O.Sauer, W.Schneider

Druck:

Ferling Druck, Darmstadt

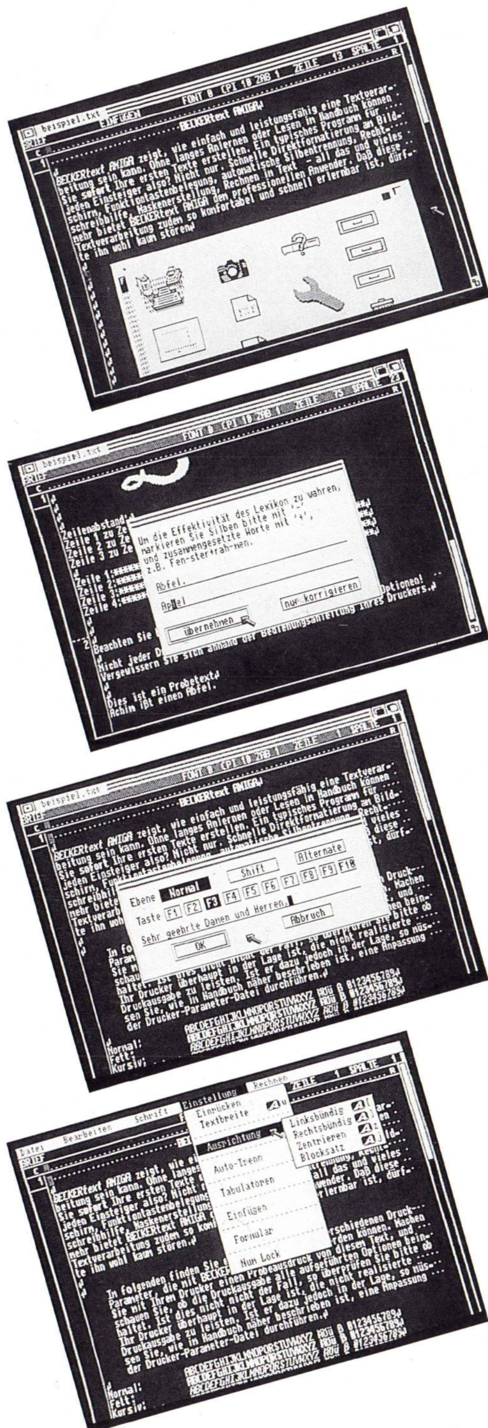
Bezugsmöglichkeit:

Zeitschriftenhandel, Kauf- und Warenhäuser, Commodore-Fachhändler oder direkt beim Verlag.

KICKSTART erscheint 11 mal im Jahr
Einzelpreis:
DM 7,- ,ÖS 56,- SFr 7,-
Jahresabonnement Inland:
DM 70,-
Europ. Ausland DM 90,-
Luftpost DM 120.-

„WORT PERFEKT"

BECKERtext Amiga



DAS DEUTSCHE TEXTWUNDER

Alle, die viel schreiben, brauchen eine Textverarbeitung, die alles kann und trotzdem schnell und komfortabel ist. Denn was nützt der größte Leistungsumfang, wenn man die Vielfalt der Funktionen nicht im Kopf hat und immer wieder das Handbuch wälzen muß?
Nein, eine Alleskönner-Textverarbeitung muß her. Mit allen Features, die man wirklich braucht, der vollen Integration in die AMIGA-INTUITION-Oberfläche – sprich: Anklicken aller Befehle mit der Maus – und dazu noch ein akzeptabler Preis. Wunschtraum oder Realität?
Die Antwort heißt BECKERtext AMIGA.

SCHNELLE DIREKTFORMATIERUNG: WYSIWYG-Prinzip: keine störenden Steuerzeichen im Text, schnelle Direktformatierung am Bildschirm mit allen Attributen (fett, kursiv, unterstrichen, Blocksatz, zentriert, linksbündig, rechtsbündig, hochstellen, tiefstellen, Horizontal- oder Vertikaldruck, Variation der Zeichendichte).

EINBINDUNG VON GRAFIKEN: Wenn schon AMIGA, dann auch eine Textverarbeitung, die Grafiken verarbeitet.
Für BECKERtext kein Problem: Das integrierte Hilfsprogramm BTSnap kann alle Grafiken im IFF-Format (Dateiformat, mit dem fast alle Mal- und Zeichenprogramme für den AMIGA arbeiten) und Bildschirmausschnitte der Workbench einlesen. Eine starke Sache.

RECHNEN IM TEXT: Eine Textverarbeitung soll souverän mit Worten operieren, aber wie ist es mit Zahlen? Für BECKERtext AMIGA eine Leichtigkeit: Rechnen im Text, sowohl spalten- als auch zeilenweise. Mit bis zu 6 Nachkommastellen und 10stelliger Genauigkeit. Selbstverständlich mit Dezimaltabulator. Ein besonderer Vorteil für die Tabellenverarbeitung.

FORMULARE NACH WAHL: Mit BECKERtext AMIGA können Sie beliebige Formulare definieren und bis auf Abruf speichern (z. B. für Rechnungen, Lastschriftformulare, Tabellen, Briefpapier, Seitenlayout, etc.). Die lästige Neudefinition bewährter Standardformate entfällt – wieder ein Pluspunkt mehr.

ELEKTRONISCHE RECHTSCHREIBHILFE: Normalerweise folgt jeder Texteingabe die Korrektur. BECKERtext AMIGA leistet Vorarbeit: Das integrierte ONLINE-Lexikon überprüft den Text schon während der Eingabe auf Fehler in der Rechtschreibung (wahlweise auch danach). Da es individuell erweiterbar ist, eignet es sich auch für Fremdsprachen.

ÜBERLEGENE FEATURES: Mehrspaltige Druckausgabe. Beim Ausdruck können Textdateien miteinander verknüpft werden. Multitasking: paralleles Arbeiten mit mehreren Programmen in verschiedenen Fenstern. Von einer Vorlage können bis zu 99 Kopien nacheinander ausgedruckt werden. Dreifache Funktionstastenbelegung mit maximal 160 Zeichen zur Speicherung von Floskeltexten oder Tastaturmakros. 1- und 2-bahniger Etikettenausdruck. Automatisches Erstellen von Stichwort- und Inhaltsverzeichnissen. Serienbrieffunktion mit Übernahmemöglichkeit aus beliebigen ASCII-Dateien. Datentransfer über RS 232. Umfangreiche Blockoperationen (Suchen, Ersetzen, Kopieren, Verschieben). Komfortable Druckeranpassung mit integriertem Treiber für alle gängigen Drucker. Querdruck auf Epson-kompatiblen Druckern bis zu 999 Zeichen pro Zeile. Ausführliches deutsches Handbuch. Minimalkonfiguration: 1 MByte RAM.

BECKERtext AMIGA nur DM 199,–

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

COUPON
COUPON BITTE EINSENDEN AN:
DATA BECKER
MEROWINGERSTR. 30
4000 DÜSSELDORF

HIERMIT BESTELLE ICH

NAME, VORNAME

STRASSE

ORT

Schlag auf Schlag

Software für Ihren AMIGA

Klassische Renner ebenso wie brand-aktuelle Programme, z. Tl. erstmalig auf dem deutschen Markt. Top-Hits zu Top-Preisen. Vergleichen Sie selbst! Wir liefern nur Originalprogramme zu knallhart kalkulierten Preisen. Nutzen Sie diese Anzeige für Ihre Bestellung: Einfach ankreuzen ☒ und ab zur Post! Oder rufen Sie uns an. Lieferung gegen Vorauskasse (Bar, Scheck, Kreditkarte, zzgl. DM 4,00 Porto) oder Nachnahme (zzgl. DM 6,00 Porto). Nur Versandhandel, kein Ladenverkauf. Lieferung, solange Vorrat reicht.
GTI GmbH, Zimmersmühlenweg 73, 6370 Oberursel
Telefon (0 61 71) 7 30 48, Telefax (0 61 71) 83 02



Programm	DM
SPIELE UND SIMULTATIONEN	
☐ Arazok's Tomb	65,00
☐ Arkanoid	69,00
☐ Backlash	54,90
☐ Barbarian	74,95
☐ Bards Tale	89,00
☐ Battleships	54,95
☐ Cogans Run	49,00
☐ Crack	59,95
☐ Crazy Cars	64,90
☐ Crystal Hammer	39,00
☐ Dark Castle	75,00
☐ Defender of the Crown	85,00
☐ Deja Vu	85,00
☐ Destroyer	74,95
☐ Ebonstar	74,95
☐ Emerald Mine	29,00
☐ Emetic Skimmer	49,00
☐ Faery Tale Adventure	85,00
☐ Feud	29,00
☐ Ferrari Foruma One	89,00
☐ Firepower	49,95
☐ Flight Simulator II	99,00
☐ F.Sim Scenery Disk # 7	49,90
☐ F.Sim Scenery Disk # 11	49,90
☐ Footman	49,95
☐ Formula One Grand Prix	59,95
☐ Frost Byte	44.95
☐ Fußball Manager	74,95
☐ Galactic Invasion	44,95
☐ Galileo 2.0	120,00
☐ Garrison	69,00
☐ Girls of Riviera	44,95
☐ Golden Path	59,00
☐ Goldrunner	75,00
☐ Gnome Ranger	44,95
☐ Guild of Thieves	64,90
☐ Great Giana Sisters	59,00
☐ Hollywood Poker	39,00
☐ Indoor Sports	74,95
☐ Insanity Fight	75,00
☐ Into the Eagles Nest	75,00
☐ In 80 Days around the World	59,00
☐ Impact	49,95
☐ Jagd auf Roter Oktober	74,95
☐ Jet	89,95
☐ Jinxter	79,00
☐ Jump Jet	44,95
☐ Kikstart 2	29,00
☐ Knight Orc	59,00
☐ Land of Lounge Lizards	90,00
☐ Mercenary	64,90
☐ Moebius	79,95
☐ Mouse Trap	44,95
☐ Obliterator	74,95
☐ Phalanx II	29,00
☐ Phantasie III	59,95
☐ Plutos	49,95
☐ Ports of Call	84,95
☐ Q-Ball	59,00
☐ Roadwars	54,90
☐ Roadwar Europa	59,95
☐ Seconds Out	54,95
☐ Shadowgate	89,00
☐ Silent Service	75,00
☐ Sindbad	85,00
☐ Slaygon	54,95
☐ Soccer Supremo	44,95
☐ Space Ranger	29,00
☐ Star Glider	69,00
☐ Strike Force Harrier	74,95
☐ Super Huey	60,00
☐ Terramex	59,00
☐ Terrorpods	74,95
☐ Test Drive	74,95
☐ Tetris	59,00
☐ Thunderboy	59,00
☐ Time Bandit	54,90
☐ Uninvited	85,00
☐ Vyper	49,95
☐ Western Games	59,00
☐ Winter Games	75,00
☐ Winter Olympiad 88	54,95
☐ World Games	75,00
☐ Xenon	54,90
☐ XR-35	29,00
SCHACHECKE	
☐ Art of Chess	64,90
☐ Großmeister	59,00
☐ Sargon III	95,00
ANIMATIONS- UND GRAFIKSOFTWARE TEXTVERARBEITUNG UND DESKTOP PUBLISHING	
☐ Aegis Animator/Images	235,00
☐ Aegis Draw Plus	445,00
☐ Aegis Video Titler v1.1	249,00
☐ Analytic Art	110,00
☐ Animate 3D	275,00
☐ Animate 3D + deutsche Anl.	310,00
☐ Butcher (Deutsch PAL)	110,00
☐ Calligrafonts	99,00
☐ Deluxe Paint II (Deutsch PAL)	249,00
☐ Digi-Droid	199,00
☐ Digipaint (Deutsch PAL)	138,00
☐ Digiview (Deutsch PAL)	440,00
☐ Digiview PAL Software	19,95
☐ Director	125,00
☐ Director + deutsche Anleitung	155,00
☐ Express Paint v2.0	185,00
☐ Forms in Flight	145,00
☐ Gender Changer	55,00
☐ Graphic Studio	105,00
☐ Interchange	85,00
☐ IntroCAD	140,00
☐ Page Flipper	85,00
☐ Photon Paint	195,00
☐ PIXmate	120,00
☐ Prism +	120,00
☐ Professional Page	660,00
☐ Sculpt 3D	190,00
☐ Silver 3D	280,00
☐ TV Show	185,00
☐ TV Text	179,00
☐ X-CAD Designer	1145,00
☐ Zuma Fonts Volume I	60,00
☐ Zuma Fonts Volume II	60,00
☐ Zuma Fonts Volume III	60,00
PROGRAMMIERSPRACHEN UND UTILITIES	
☐ AC Basic	360,00
☐ AC Fortran	545,00
☐ Aztec C 3.6 (DEV)	595,00
☐ Aztec C 3.6 (PROF)	389,00
☐ Disk Master	119,00
☐ FACC II – Floppy Accelerator	59,95
☐ GOMF 2.0	59,95
☐ Go – 64	139,90
☐ Grabbit	54,00
☐ Inovatools # 1	140,00
☐ Intswitch	27,50
☐ Lattice C 4.0	385,00
☐ LV Backup	120,00
☐ M2 Amiga (Deutsch)	350,00
☐ Marauder II (mit Brainfile 10)	69,00
☐ Metacomco Assembler	185,00
☐ Metacomco Pascal	185,00
☐ Metacomco Shell	135,00
☐ Modula 2 (Regular)	185,00
☐ Modula 2 (Developers)	275,00
☐ Modula 2 (Commercial)	545,00
☐ Quarterback	135,00
☐ The 64 Emulator	148,00
☐ True Basic	195,00
☐ Turbo Print	89,00
☐ Virus Finder	49,00
MUSIKPROGRAMME	
☐ ADRUM	99,00
☐ Aegis Audiomaster	99,00
☐ Aegis Audiomaster + dtsche. Anl.	134,00
☐ Aegis Sonix	135,00
☐ Casio CZ Editor/Librarian	225,00
☐ Deluxe Music Construction Set	159,00
☐ Deluxe Music Constr. Set (Deutsche PAL)	199,00
☐ Dynamic Drums	135,00
☐ Dynamic Studio	375,00
☐ DX7 Master Editor/Librarian	275,00
☐ D50 Master Editor/Librarian	275,00
☐ ECE MIDI Interface	130,00
☐ Generic Editor/Librarian	225,00
☐ Hotlicks	89,00
☐ Instant Music	85,00
☐ Keyboard Controlled Sequencer vs1.6	445,00
☐ MIDI Gold (für Amiga 1000)	170,00
☐ Music Mouse	125,00
☐ Pro MIDI Studio V 1.4	345,00
☐ Soundsampler	225,00
DATENFERNÜBERTRAGUNG	
☐ Aegis Diga	135,00
BUSINESSPROGRAMME	
☐ Acquisition 1.3F	545,00
☐ Aegis Impact	150,00
☐ Haicalc	105,00
☐ Logistix (Deutsch)	340,00
☐ Maxiplan 500	249,00
☐ Maxiplan Plus	345,00
BÜCHER UND ZEITSCHRIFTEN	
☐ Amazing Computing (Porto frei)	12,00
☐ Flying Flight Simulator	19,95
☐ Music Through MIDI	39,95
☐ Robo City News	4,95
DISKETTEN	
☐ Rhone-Poulenc 3,5" DS/DD	Preise a. A.
VERSCHIEDENES	
☐ Flicker Master	35,00
☐ Megacover (PVC-Haube für A500 + Maus)	29,95
☐ Mouse House	19,90
☐ Mouse Mat	16,50

GTI. Spezialist für AMIGA-Software

